INTERNATIONAL STANDARD

ISO/IEC 29500-1

Fourth edition
2016-11-01

Information technology — Document description and processing languages — Office Open XML File Formats —

Part 1: Fundamentals and Markup Language Reference

Technologies de l'information — Description des documents et langages de traitement — Formats de fichier "Office Open XML" —

Partie 1: Principes essentiels et référence de langage de balisage

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Reference number
ISO/IEC 29500-1:2016(E)

© ISO/IEC 2016

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

 © ISO/IEC 2016 – All rights reserved

Table of Contents

Foreword ..... **viii**

Introduction ..... **x**

1. Scope **1**

2. Conformance **2**

2.1 Document Conformance 2
2.2 Application Conformance 2
2.3 Application Descriptions 3
2.4 Interoperability Guidelines 5

3. Normative References **6**

4. Terms and Definitions **10**

5. Notational Conventions **13**

6. Acronyms and Abbreviations **14**

7. General Description **15**

8. Overview **16**

8.1 Content Overview 16
8.2 Packages and Parts 16
8.3 Consumers and Producers 16
8.4 WordprocessingML 16
8.5 SpreadsheetML 18
8.6 PresentationML 18
8.7 Supporting MLs 19

9. Packages **21**

9.1 Office Open XML's Use of OPC 21
9.2 Relationships in Office Open XML 22

10. Markup Compatibility and Extensibility **27**

11. WordprocessingML **28**

11.1 Glossary of WordprocessingML-Specific Terms 28
11.2 Package Structure 28
11.3 Part Summary 31
11.4 Document Template 57
11.5 Framesets 58
11.6 Master Documents and Subdocuments 59
11.7 Mail Merge Data Source 60
11.8 Mail Merge Header Data Source 61
11.9 XSL Transformation 62

12. SpreadsheetML **64**

12.1 Glossary of SpreadsheetML-Specific Terms 64
12.2 Package Structure 65

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

12.3 Part Summary 67
12.4 External Workbooks 102

13. PresentationML 103

13.1 Glossary of PresentationML-Specific Terms 103
13.2 Package Structure 103
13.3 Part Summary 106
13.4 HTML Publish Location 124
13.5 Slide Synchronization Server Location 125

14. DrawingML 127

14.1 Glossary of DrawingML-Specific Terms 127
14.2 Part Summary 127

15. Shared 140

15.1 Glossary of Shared Terms 140
15.2 Part Summary 141
15.3 Hyperlinks 162

16. Part Overview 164

16.1 WordprocessingML Summary 164
16.2 SpreadsheetML Summary 164
16.3 PresentationML Summary 165
16.4 DrawingML Summary 166
16.5 Shared Summary 166

17. WordprocessingML Reference Material 169

17.1 Table of Contents 169
17.2 Main Document Story 189
17.3 Paragraphs and Rich Formatting 195
17.4 Tables 373
17.5 Custom Markup 485
17.6 Sections 547
17.7 Styles 614
17.8 Fonts 670
17.9 Numbering 692
17.10 Headers and Footers 734
17.11 Footnotes and Endnotes 747
17.12 Glossary Document 780
17.13 Annotations 798
17.14 Mail Merge 929
17.15 Settings 970
17.16 Fields and Hyperlinks 1158
17.17 Miscellaneous Topics 1293
17.18 Simple Types 1302

18. SpreadsheetML Reference Material 1523

18.1 Table of Contents 1523
18.2 Workbook 1542
18.3 Worksheets 1589

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.4 Shared String Table 1717
18.5 Tables 1726
18.6 Calculation Chain 1742
18.7 Comments 1745
18.8 Styles 1752
18.9 Metadata 1801
18.10 Pivot Tables 1815
18.11 Shared Workbook Data 1959
18.12 QueryTable Data 1988
18.13 External Data Connections 1995
18.14 Supplementary Workbook Data 2016
18.15 Volatile Dependencies 2026
18.16 Custom XML Mappings 2031
18.17 Formulas 2039
18.18 Simple Types 2434

19. PresentationML Reference Material 2516
19.1 Table of Contents 2516
19.2 Presentation 2522
19.3 Slides 2559
19.4 Comments 2597
19.5 Animation 2601
19.6 Slide Synchronization Data 2690
19.7 Simple Types 2691

20. DrawingML - Framework Reference Material 2719
20.1 DrawingML - Main 2719
20.2 DrawingML - Picture 3087
20.3 DrawingML - Locked Canvas 3095
20.4 DrawingML - WordprocessingML Drawing 3096
20.5 DrawingML - SpreadsheetML Drawing 3152

21. DrawingML - Components Reference Material 3181
21.1 DrawingML - Main 3181
21.2 DrawingML - Charts 3361
21.3 DrawingML - Chart Drawings 3469
21.4 DrawingML - Diagrams 3490

22. Shared MLs Reference Material 3599
22.1 Math 3599
22.2 Extended Properties 3720
22.3 Custom Properties 3727
22.4 Variant Types 3729
22.5 Custom XML Data Properties 3740
22.6 Bibliography 3743
22.7 Additional Characteristics 3780
22.8 Office Document Relationships 3784
22.9 Shared Simple Types 3785

23. Custom XML Schema References 3800

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

23.1 Table of Contents 3800
23.2 Elements 3800

Annex A. (normative) Schemas – W3C XML Schema 3805
A.1 WordprocessingML 3805
A.2 SpreadsheetML 3871
A.3 PresentationML 3955
A.4 DrawingML - Framework 3986
A.5 DrawingML - Components 4054
A.6 Shared MLs 4105
A.7 Custom XML Schema References 4129

Annex B. (informative) Schemas – RELAX NG 4131
B.1 WordprocessingML 4132
B.2 SpreadsheetML 4178
B.3 PresentationML 4273
B.4 DrawingML - Framework 4298
B.5 DrawingML - Components 4350
B.6 Shared MLs 4386
B.7 Custom XML Schema References 4402
B.8 Additional Resources 4403

Annex C. (informative) Additional Syntax Constraints 4405

Annex D. (informative) Namespace Prefix Mapping in Examples 4406

Annex E. (informative) WordprocessingML Custom XML Data Extraction 4408

Annex F. (normative) WordprocessingML Page Borders 4411

Annex G. (normative) Predefined SpreadsheetML Style Definitions 4412
G.1 Built-in Table Styles 4412
G.2 Built-in Cell Styles 4468
G.3 Built-in PivotTable AutoFormats 4472

Annex H. (informative) Example Predefined DrawingML Shape and Text Geometries 4488

Annex I. (informative) Bidirectional Support 4489
I.1 Introduction 4489
I.2 Shared (WordprocessingML and DrawingML) 4489
I.3 WordprocessingML 4491
I.4 SpreadsheetML 4494
I.5 PresentationML 4495
I.6 DrawingML 4495
I.7 The Unicode Bidirectional Algorithm and Office Open XML 4495

Annex J. (informative) Accessibility Best Practices 4499
J.1 The Value of Creating an Accessible Office Open XML Implementation 4499
J.2 Needs by Type of Disability 4500
J.3 Best Practices for Developers 4503
J.4 Best Practices for Document and Template Authors 4506
J.5 Best Practices for Customers of Office Open XML Implementations 4519

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Annex K. (informative) Root Element Locations 4532
K.1 Grouped by Part Name 4532
K.2 Grouped by Schema Name 4534
Annex L. (informative) Primer 4538
L.1 Introduction to WordprocessingML 4538
L.2 Introduction to SpreadsheetML 4630
L.3 Introduction to PresentationML 4769
L.4 Introduction to DrawingML 4801
L.5 Introduction to VML 4960
L.6 Introduction to Shared MLs 4974
L.7 Miscellaneous Topics 5005
Annex M. (informative) Differences Between ISO/IEC 29500 and ECMA-376:2006 5016
M.1 WordprocessingML 5016
M.2 SpreadsheetML 5019
M.3 PresentationML 5020
M.4 DrawingML 5021
M.5 VML 5022
M.6 Shared 5022
M.7 Custom XML Schema References 5023
Bibliography 5024

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29500-1 was prepared by ISO/IEC JTC 1, Information technology, Subcommittee SC 34, Document description and processing languages.

This fourth edition cancels and replaces the third edition (ISO/IEC 29500-1:2012), which has been technically revised by incorporation of the Technical Corrigenda ISO/IEC 29500-1:2012/Cor.1:2015 and ISO/IEC 29500-1:2012/Cor.2:2016.

ISO/IEC 29500 consists of the following parts, under the general title *Information technology — Document description and processing languages — Office Open XML File Formats*:

- *Part 1: Fundamentals and Markup Language Reference*
- *Part 2: Open Packaging Conventions*
- *Part 3: Markup Compatibility and Extensibility*
- *Part 4: Transitional Migration Features*

Annexes A, F and G form a normative part of this Part of ISO/IEC 29500. Annexes B–E and H–M are for information only.

This Part of ISO/IEC 29500 includes five annexes (Annex A, Annex B, Annex F, Annex G, and Annex H) that refer to data files provided in electronic form.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The document representation formats defined by this Part are different from the formats defined in the corresponding Part of ECMA-376:2006. Some of the differences are reflected in schema changes, as shown in Annex M of this Part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Introduction

ISO/IEC 29500 specifies a family of XML schemas, collectively called *Office Open XML*, which define the XML vocabularies for word-processing, spreadsheet, and presentation documents, as well as the packaging of documents that conform to these schemas.

The goal is to enable the implementation of the Office Open XML formats by the widest set of tools and platforms, fostering interoperability across office productivity applications and line-of-business systems, as well as to support and strengthen document archival and preservation, all in a way that is fully compatible with the existing corpus of Microsoft Office documents.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Information technology — Document description and processing languages — Office Open XML File Formats

Part 1:

Fundamentals and Markup Language Reference

1. Scope

ISO/IEC 29500 defines a set of XML vocabularies for representing word-processing documents, spreadsheets and presentations. On the one hand, the goal of ISO/IEC 29500 is to be capable of faithfully representing the pre-existing corpus of word-processing documents, spreadsheets and presentations that had been produced by the Microsoft Office applications (from Microsoft Office 97 to Microsoft Office 2008, inclusive) at the date of the creation of ISO/IEC 29500. It also specifies requirements for Office Open XML consumers and producers. On the other hand, the goal is to facilitate extensibility and interoperability by enabling implementations by multiple vendors and on multiple platforms.

This Part of ISO/IEC 29500 specifies concepts for documents and applications of both strict and transitional conformance.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

2. Conformance

2.1 Document Conformance

A document of conformance class Office Open XML Strict shall be a package of conformance class OPC, as specified in ISO/IEC 29500-2, for which all the following shall hold:

- The document obeys all constraints specified in this Part of ISO/IEC 29500
- The document is of category Wordprocessing, Spreadsheet, or Presentation, as defined in §4
- For each OPC Part of the document of the types listed in §11.3, §12.3, §13.3, §14.2 or §15.2, all the following shall hold:
    i. The Part may contain markup in the Markup Compatibility namespace as specified in ISO/IEC 29500-3
    ii. After the removal of any extensions by an MCE processor as specified in ISO/IEC 29500-3, the part is valid against the strict W3C XML Schema (Appendix A)

This Part of ISO/IEC 29500 uses the following further terms to refer to documents of conformance class Office Open XML Strict:

- *WML Strict*, if the document is of category Wordprocessing
- *SML Strict*, if the document is of category Spreadsheet
- *PML Strict*, if the document is of category Presentation

2.2 Application Conformance

Application conformance incorporates both syntax and semantics:

- A conforming consumer shall not reject any conforming documents of at least one document conformance class.
- A conforming producer shall be able to produce conforming documents of at least one document conformance class.
- A conforming application shall treat the information in Office Open XML documents in a manner consistent with the semantic definitions given in ISO/IEC 29500. An application's intended behavior need not require that application to process all of the information in an Office Open XML document. However, the information that it does process shall be processed in a manner that is consistent with the semantic definitions given in ISO/IEC 29500.

[*Note*: This note illustrates the third bullet above. Conforming applications might serve various functions. Examples include a viewer, an editor, and a back-end processor. Here is an illustration of how the third bullet applies to each of those examples:

- If a conforming viewer supports a given feature, then when it displays information using that feature, it respects the semantics of that feature as described in the Standard.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- If a conforming editor supports a given feature, then when it provides its user with an interface for manipulating information using that feature, it respects the semantics of that feature as described in the Standard.
- If a conforming back-end processor supports a given feature, then when that processor transforms or assembles information involving that feature, that processor respects the semantics of that feature as described in the Standard.

end note]

This Part of ISO/IEC 29500 defines the following application conformance classes:

- *WML Strict*, if the application is a conforming application that is a consumer or producer of documents having conformance class WML Strict.
- *SML Strict*, if the application is a conforming application that is a consumer or producer of documents having conformance class SML Strict.
- *PML Strict*, if the application is a conforming application that is a consumer or producer of documents having conformance class PML Strict.

Conformance can also involve the use of application descriptions; see §2.3 for details.

2.3 Application Descriptions

An application can be defined as conforming to zero or more *application descriptions* in a particular conformance class.

The application descriptions defined within ISO/IEC 29500 are:

- Base
- Full

[*Note*: These application descriptions should not be taken as limiting the ability of an application provider to create innovative applications. They are intended as a mechanism for labelling applications rather than for restricting their capabilities. The intention is to promote interoperability between different applications that share the same conformance class. Application descriptions are orthogonal to the conformance of the documents produced by those applications. For example, a tool used for automated translation of documents might have an application description of “Base” but will still produce fully conformant documents. *end note*]

The application descriptions are determined in terms of an application’s semantic understanding of particular features. *Semantic understanding* is to be interpreted in that an application shall treat the information in Office Open XML documents in a manner consistent with the semantic definitions given in ISO/IEC 29500.

Each application description is identified by a URI.

The application descriptions are defined in the following subclauses.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

2.3.1 Base Application Description

Description URI: http://purl.oclc.org/ooxml/descriptions/base

An application conforming to this description has a semantic understanding of at least one feature within its conformance class.

[*Note*: In addition, applications that include a user interface are strongly recommended to support all accessibility features appropriate to that user interface. *end note*]

2.3.2 Full Application Description

Description URI: http://purl.oclc.org/ooxml/descriptions/full

An application conforming to this description has a semantic understanding of every feature within its conformance class.

2.3.3 Additional Application Descriptions

It is expected that additional application descriptions will be defined within the maintenance process for ISO/IEC 29500. It is also expected that third parties might define their own application descriptions; for example to inform their procurement decisions, or to deal with domains such as accessibility.

[*Note*: A possible application description would be a "standard" application description for a wordprocessing application. This could be created by taking the intersection of the features available in common wordprocessing applications such as Word 2000, OpenOffice 2, WordPerfect, and iWork Pages. In addition, it could define formats such as specific image and video formats required to be supported to conform to the description. Similar descriptions could be created for spreadsheet applications and presentation applications. Such a description would promote interoperability between applications implementing OOXML. It would also promote interoperability between applications implementing OOXML and applications implementing other document formats such as ISO/IEC 26300. *end note*]

Application descriptions are not required to be strict subsets of each other. An application can simultaneously conform to multiple application descriptions.

Any such newly created description shall enumerate the features that are required for conformance to it. Such a description should provide a machine-processable schema, preferably using a standard such as ISO/IEC 19757.

[*Note*: If the application conforming to a description is a document consumer, it should be able to consume any document that respects such a schema associated with the description. If the application is a document producer, any document produced by that application should respect the schema of the description. *end note*]

Any such description should be identified using a URI, in a similar manner to the names used for application descriptions within ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: For the convenience of users of the description, it is recommended that creators of a description should make a human- or machine-readable form of that description available at a URL corresponding to the description URI. *end note*]

2.3.4 Representation of Application Descriptions within Documents

An application description is related to applications, rather than to document conformance. Therefore, there is no normative mechanism for representing an application description within a document.

[*Note*: It is recommended that implementers wishing to represent an application description within a document use the standard metadata mechanism for Office Open XML. *end note*]

2.4 Interoperability Guidelines

[*Guidance*: The following interoperability guidelines incorporate semantics.

For the guidelines to be meaningful, a software application should be accompanied by documentation that describes what subset of ISO/IEC 29500 it supports. The documentation should highlight any behaviors that would, without that documentation, appear to violate the semantics of document XML elements. Together, the application and documentation should satisfy the following conditions.

1. The application need not implement operations on all XML elements defined in ISO/IEC 29500. However, if it does implement an operation on a given XML element, then that operation should use semantics for that XML element that are consistent with ISO/IEC 29500.
2. If the application moves, adds, modifies, or removes XML element instances with the effect of altering document semantics, it should declare the behavior in its documentation.

The following scenarios illustrate these guidelines.

- A presentation editor that interprets the preset shape geometry "`rect`" as an ellipse does not observe the first guideline because it implements "`rect`" but with incorrect semantics.
- A batch spreadsheet processor that saves only computed values even if the originally consumed cells contain formulas, might satisfy the first condition, but does not observe the second because the editability of the formulas is part of the cells' semantics. To observe the second guideline, its documentation should describe the behavior.
- A batch tool that reads a word-processing document and reverses the order of text characters in every paragraph with "Title" style before saving it can be conforming even though ISO/IEC 29500 does not recommend this behavior. This tool's behavior would be to transform the title "Office Open XML" into "LMX nepO eciffO". Its documentation should declare its effect on such paragraphs.

The normative requirements in §2.1 imply that a conforming producer shall not write unescaped non-XML characters. As an implementation guideline, a conforming producer additionally should not write escaped non-XML characters. Doing so damages interoperability with existing XML-based standards such as SOAP and RDF. For example, implementers could either refuse to create documents including such characters, or warn users that including such characters compromises the re-usability of their documents. *end guidance*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

3. Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ANSI X3.4-1986, *American Standard Code for Information Interchange (ASCII)*

Bureau of Standards, Metrology and Inspection of the Ministry of Economic Affairs, *CNS 7648: Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times*

Calendar Reform Committee, Indian Ephemeris and Nautical Almanac. 1957

Stokes, M., M. Anderson, S. Chandrasekar, and R. Motta. *A Standard Default Color Space for the Internet. Vers. 1.10*. November 5, 1996. http://www.w3.org/Graphics/Color/sRGB

Har'El, Zvi, *Gauss Formula for the Julian Date of Passover*. Deptartment of Mathematics, Technion, Israel Institue of Technology, Haifa 32000, Isreal, 2005, 6

Duerst, M, and M Suignard. *Internationalized Resource Identifiers (IRIs).* IETF. January 2005. http://tools.ietf.org/html/rfc3987

IANA, *Character Sets from IANA*, as specified at http://www.iana.org/assignments/character-sets

IANA. *MIME Media Types*. Internet Assigned Numbers Authority. http://www.iana.org/assignments/media-types/

IEC 60559:1989, *Binary Floating-Point Arithmetic for Microprocessor Systems*

ISO/IEC 2382-1:1993, *Information technology — Vocabulary — Part 1: Fundamental terms*

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1* (referred to in ISO/IEC 29500 as the ANSI character set)

ISO/IEC 9075-1, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

ISO/IEC 9075-3, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)*

ISO/IEC 9945-2, *Information technology — Portable Operating System Interface (POSIX) — Part 2: System Interfaces*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*.

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*.

ISO/IEC 14496-22:2009, *Information technology — Coding of audio-visual objects — Part 22: Open Font Format*

ISO/IEC 29500-2:2012, Information technology — Document description and processing languages — Office Open XML File Formats – Part 2: Open Packaging Conventions

ISO/IEC 29500-3:2015, Information technology — Document description and processing languages — Office Open XML File Formats – Part 3: Markup Compatibility and Extensibility

Japanese Industrial Standard, JIS X 0301: *Data elements and interchange formats —Information interchange — Representation of dates and times*. Japan, 2002.

Kingdom of Saudi Arabia, Ministry of Islamic Affairs, Endowments, Da'wah and Guidance.

Korean Law Enactment No. 4, 1961.

Faure, D. (n.d.). *Creating and Using Components (KParts)*. http://techbase.kde.org/Projects/Documentation.

Maimon, Rabbi Moshe ben, *Complete Restatement of the Oral Law (Mishneh Torah)*.

Ausbrooks, Ron, et al. *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*. October 21, 2003. http://www.w3.org/TR/MathML/.

Kaliski, B. *The MD2 Message-Digest Algorithm*. April 1992. http://www.ietf.org/rfc/rfc1319.txt

Rivest, R. *The MD4 Message-Digest Algorithm*. April 1992. http://www.ietf.org/rfc/rfc1320.txt

The MD5 Message-Digest Algorithm. April 1992. http://www.ietf.org/rfc/rfc1321.txt.

National Measurement Regulations 1999, Commonwealth of Australia
http://www.comlaw.gov.au/Details/F2011C00445

NIST Guide to SI Units, http://physics.nist.gov/Pubs/SP811/appenB9.html

QuickTime File Format Specification (2007-09-04 version)
http://developer.apple.com/standards/classicquicktime.html

Resource Description Framework (RDF), http://www.w3.org/RDF/

RFC 822, *Standard for ARPA Internet Text Messages* (http://www.ietf.org/rfc/rfc0822.txt)

RFC 2045, Borenstein, N., and N. Freed. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. The Internet Society. 1996. http://www.ietf.org/rfc/rfc2045.txt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

RFC 2119, Bradner, Scott, 1997: *Key words for use in RFCs to Indicate Requirement Levels*. http://www.ietf.org/rfc/rfc2119.txt

RFC 2616, Berners-Lee, T., R. Fielding, H. Frystyk, J. Gettys, P. Leach, L. Masinter, and J. Mogul. *Hypertext Transfer Protocol—HTTP/1.1*. The Internet Society. 1999. http://www.ietf.org/rfc/rfc2616.txt

RFC 3066, Alvestrand, H. *Tags for the Identification of Languages*. The Internet Society. 2001. http://www.ietf.org/rfc/rfc3066.txt

RFC 3339, Klyne, G. and C. Newman. *Date and Time on the Internet: Timestamps*. The Internet Society. 2002. http://www.ietf.org/rfc/rfc3339.txt

RFC 3629, Yergeau, F. *UTF-8, a transformation format of ISO 10646*. The Internet Society. 2003. http://www.ietf.org/rfc/rfc3629.txt

RFC 3986, Berners-Lee, T., R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. The Internet Society. 2005. http://www.ietf.org/rfc/rfc3986.txt

Simple Object Access Protocol (SOAP), http://www.w3.org/TR/soap12

SMIL, Bulterman, D., Grassel, G., Jansen, J., Koivisto, A., Layaïda, N., Michel, T., et al. (2005, December 13). *Synchronized Multimedia Integration Language (SMIL 2.1)*. Retrieved from W3C: http://www.w3.org/TR/SMIL/

SVG, Andersson, O., Armstrong, P., Axelsson, H., Berjon, R., Bézaire, B., Bowler, J., et al. (2003, January 14). *Scalable Vector Graphics (SVG) 1.1 Specification*. Retrieved from W3C - World Wide Web Consortium: http://www.w3.org/TR/SVG/

The GNOME Project. (2003, December 12). *Component Model - Bonobo Document Model*. Retrieved from The GNOME Development Site: http://developer.gnome.org/bonobo-activation/stable/

The Unicode Consortium. *The Unicode Standard*, http://www.unicode.org/standard/standard.html.

Unicode Technical Note #28, *Nearly Plain-Text Encoding of Mathematics*. August 29, 2006, http://www.unicode.org/notes/tn28

United States Postal Service. *Domestic Mail Manual*. United States Postal Service. November 8, 2007. http://pe.usps.com/cpim/ftp/manuals/dmm300/Full/MailingStandards.pdf

The Units of Measurement Regulations 1995, United Kingdom http://www.opsi.gov.uk/si/si1995/Uksi_19951804_en_2.htm

Universal Postal Union. *POST*CODE: Postal addressing systems*. Berne: UPU Publications, 2006, ISBN 92-95025-37-7, ISSN 1020-6019

Web Accessibility Initiative (WAI), http://www.w3.org/WAI/

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

XSLT, Clark, James, *XSL Transformations (XSLT) Version 1.0*, World Wide Web Consortium Recommendation. 1999. http://www.w3.org/TR/xslt

XML, Tim Bray, Jean Paoli, Eve Maler, C. M. Sperberg-McQueen, and François Yergeau (editors). *Extensible Markup Language (XML) 1.0, Fourth Edition*. World Wide Web Consortium. 2006. http://www.w3.org/TR/2006/REC-xml-20060816/ [Implementers should be aware that a further correction of the normative reference to XML to refer to the 5th Edition will be necessary when the related Reference Specifications to which this International Standard also makes normative reference and which also depend upon XML, such as XSLT, XML Namespaces and XML Base, are all aligned with the 5th Edition.]

XML Base, Marsh, Jonathan. *XML Base*. World Wide Web Consortium. 2001. http://www.w3.org/TR/2001/REC-xmlbase-20010627/

XML Namespaces, Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin (editors). *Namespaces in XML 1.0 (Third Edition)*, 8 December 2009. World Wide Web Consortium. http://www.w3.org/TR/2009/REC-xml-names-20091208/

XPATH, Clark, James; DeRose, Steve *XML Path Language (XPath) Version 1.0*, World Wide Web Consortium Recommendation. 1999. http://www.w3.org/TR/xpath.

XML Schema Part 0: Primer (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes (Second Edition), W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-2/

.ZIP File Format Specification from PKWARE, Inc., version 6.2.0 (2004), as specified in http://www.pkware.com/documents/APPNOTE/APPNOTE_6.2.0.txt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

4. Terms and Definitions

For the purposes of this document, the following terms and definitions apply. Other terms are defined where they appear in *italic* typeface, on the left side of a syntax rule, or within subclauses of language-specific grammars (§17.16 and §18.17). Terms explicitly defined in this Part of ISO/IEC 29500 are not to be presumed to refer implicitly to similar terms defined elsewhere. [*Note*: This Part uses OPC-related terms, which are defined in ISO/IEC 29500-2. *end note*]

application — A consumer or producer.

behavior — External appearance or action.

behavior, implementation-defined — Unspecified behavior where each implementation is expected to document that behavior, which would thereby promote predictability and reproducibility within any given implementation. (This term is sometimes called “application-defined behavior”.)

behavior, locale-specific — Behavior that depends on local conventions of nationality, culture, and language.

behavior, unspecified —Behavior where ISO/IEC 29500 makes no recommendations. (This term is sometimes called “application-dependent behavior”.) [*Note*: To add an extension, an implementer must use the extensibility mechanisms described by ISO/IEC 29500 rather than trying to do so by giving meaning to otherwise unspecified behavior. *end note*]

byte — A sequence of 8 bits treated as a unit.

comment — A note attached to content in a document. Although a consumer might choose to display comments, they are not considered part of the body of the document. A comment might include the text of the note, the comment author’s name and initials, and date of creation, among other things.

consumer — A piece of software or a device that reads packages through a package implementer. A consumer is often designed to consume packages only for a specific physical package format.

content type — Describes the content stored in a part. Content types define a media type, a subtype, and an optional set of parameters, as defined in RFC 2616.

document category — One of the three categories of Office Open XML documents: Wordprocessing, Spreadsheet, and Presentation, defined as follows:

- A document whose package-relationship item contains a relationship to a Main Document part (§11.3.10) is a document of category Wordprocessing.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- A document whose package-relationship item contains a relationship to a Workbook part (§12.3.23) is a document of category Spreadsheet.
- A document whose package-relationship item contains a relationship to a Presentation part (§13.3.6) is a document of category Presentation.

An Office Open XML document can contain one or more embedded Office Open XML packages (§15.2.11) with each embedded package having any of the three document categories. However, the presence of these embedded packages does not change the category of the document.

DrawingML — A set of conventions for specifying the location and appearance of drawing elements in an Office Open XML document.

extension — Any XML element, XML attribute, relationship, or part not explicitly included in ISO/IEC 29500, but that uses the extensibility mechanisms described by ISO/IEC 29500.

id — In some XML-related technologies, the term *id* implies use of the xsd:ID data type. In this international standard, this term is used to refer to a variety of different identification schemes. See *unique identifier*.

ODBC – An implementation of ISO/IEC 9075-3:2008 “Information technology -- Database languages -- SQL – Part 3: Call-Level Interface (SQL/CLI)” or SQL/CLI-based database connectivity API. An example of a broadly used SQL/CLI-based database connectivity API is the Open Database Connectivity (ODBC) API.

Office Open XML document — A rendition of a data stream formatted using the wordprocessing, spreadsheet, or presentation ML and its related MLs as described in ISO/IEC 29500-1 and ISO/IEC 29500-4. Such a document is represented as a package as described in ISO/IEC 29500-2.

OLE – OLE in this context does not refer to any specific technology; instead, it refers to the generalized abstraction of embedding and linking objects within a document.

package— A ZIP archive that conforms to the Open Packaging Conventions specification defined in ISO/IEC 29500-2.

package, embedded— A package that has been stored as the target of an Embedded Package relationship (§15.2.11) in an Office Open XML document

PresentationML — A set of conventions for representing an Office Open XML document of category Presentation.

producer — A piece of software or a device that writes packages through a package implementer. A producer is often designed to produce packages according to a particular physical package format specification.

relationship —The kind of connection between a source part and a target part in a package. Relationships make the connections between parts directly discoverable without looking at the content in the parts, and without altering the parts themselves. (See also Package Relationships.)

relationships part — A part containing an XML representation of relationships.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

relationship, explicit — A relationship in which a resource is referenced from a source part's XML using the Id attribute of a Relationship tag.

relationship, implicit — A relationship that is not explicit.

SpreadsheetML — A set of conventions for representing an Office Open XML document of category Spreadsheet.

unique identifier — In some XML-related technologies, the term *unique identifier* implies use of the xsd:ID data type. In this international standard, this term is used to refer to a variety of different identification schemes. See *id*.

WordprocessingML — A set of conventions for representing an Office Open XML document of category Wordprocessing.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

5. Notational Conventions

The following typographical conventions are used in ISO/IEC 29500:

- The first occurrence of a new term is written in italics. [*Example*: The text in ISO/IEC 29500 is divided into *normative* and *informative* categories. *end example*]
- In each definition of a term in §4 (Terms and Definitions), the term is written in bold. [*Example*: **behavior** — External appearance or action. *end example*]
- The tag name of an XML element is written using a distinct style and typeface. [*Example*: The bookmarkStart and bookmarkEnd elements specify … *end example*]
- The name of an XML attribute is written using a distinct style and typeface. [*Example*: The dropCap attribute specifies … *end example*]
- The value of an XML attribute is written using a constant-width style. [*Example*: The attribute value of auto specifies … *end example*]
- The qualified or unqualified name of a simple type, complex type, or base datatype is written using a distinct style and typeface. [*Example*: The possible values for this attribute are defined by the ST_HexColor simple type. *end example*]

When assigned namespaces are used in examples, they are included at the beginning of the example, but with the specific namespace replaced with ellipsis ("…") for brevity.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

6. Acronyms and Abbreviations

This clause is informative

The following acronyms and abbreviations are used throughout ISO/IEC 29500:

IEC — the International Electrotechnical Commission

ISO — the International Organization for Standardization

W3C — World Wide Web Consortium

End of informative text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

7. General Description

This Part of ISO/IEC 29500 is divided into the following subdivisions:

1. Front matter (clauses 1–7);
2. Overview (clause 8);
3. Package Part Structure (clauses 9–16);
4. Reference Material (clauses 17–23);
5. Annexes

Examples are provided to illustrate possible forms of the constructions described. References are used to refer to related clauses. Notes are provided to give advice or guidance to implementers or programmers. Rationale provides explanatory material as to why something is or is not in ISO/IEC 29500. Annexes provide additional information or summarize the information contained in ISO/IEC 29500.

Clauses 1–5, 7, 9–15, 17–23, Annex A, Annex F, and Annex G form the normative part of this Part of ISO/IEC 29500; the Introduction, clauses 6, 8, and 16, Annex B–Annex E, Annex H–Annex M, as well as notes, examples, rationale, guidance, and the index, are informative.

Except for whole clauses or annexes that are identified as being informative, informative text that is contained within normative text is indicated in the following ways:

1. [*Example:* code fragment, possibly with some narrative … *end example*]
2. [*Note:* narrative … *end note*]
3. [*Rationale:* narrative … *end rationale*]
4. [*Guidance*: narrative … *end guidance*]

In addition to the declarations in the “General Description”, each annex that is informative, also contains the following text at the beginning of the annex: "**This annex is informative.**"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

8. Overview

This clause is informative.

This clause contains an overview of Office Open XML.

8.1 Content Overview

This standard contains predominantly the following three types of information:

1. Normative W3C XML Schemas, informative RELAX NG schemas and an associated validation procedure for validating document syntax against those schemas (Annex A and Annex B)
2. Descriptions of XML element semantics. The semantics of an XML element refers to its intended interpretation by a human being (chiefly in §11, §12, §13, and §14)
3. Additional syntax constraints in written form

8.2 Packages and Parts

An Office Open XML document is represented as a series of related *parts* that are stored in a container called a *package*. Information about the *relationships* between a package and its parts is stored in the package's *package-relationship ZIP item*. Information about the *relationships* between two parts is stored in the *part-relationship ZIP item* for the source part. A package is an ordinary ZIP archive, which contains that package's content-type item, relationship items, and parts. (Packages are discussed further in ISO/IEC 29500-2.)

A WordprocessingML document contains a part for the body of the text; it might also contain a part for an image referenced by that text, and parts defining document characteristics, styles, and fonts. A SpreadsheetML document contains a separate part for each worksheet; it might also contain parts for images. A PresentationML document contains a separate part for each slide.

8.3 Consumers and Producers

A tool that can read and understand a package is called a *consumer*, while one that can create a package is called a *producer*. An application can be a consumer, a producer, or both. For example, when a word processor creates a new document, it acts as a producer. When it is used to open an existing document for reading or search purposes, it acts as a consumer. When it is used to open an existing document, edit it, and save the result, it acts as both consumer and producer. Similar scenarios exist for spreadsheet and presentation applications.

8.4 WordprocessingML

This subclause introduces the overall form of a WordprocessingML package, and identifies some of its main components. (See Annex L for a more detailed introduction.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A WordprocessingML package has a relationship of type officeDocument, which specifies the location of the main part in the package. For a WordprocessingML document, that part contains the main text of the document.

A WordprocessingML package's main part starts with a word processing root element. That element contains a *body*, which, in turn, contains one or more *paragraphs* (as well as tables, pictures, and the like). A paragraph contains one or more runs, where a *run* is a container for one or more pieces of *text* having the same set of properties. Like many elements that defined a logical piece of a word processing document, each run and paragraph can have associated with it a set of *properties*. For example, a run might have the property bold, which indicates that run's text is to be displayed in a bold typeface.

A WordprocessingML document is organized into *sections*, and the layout of a page on which the text appears within a section is controlled by that section's properties. For example, each section can have its own *headers* and *footers*.

One relationship from the document part specifies the document's styles. A *style* defines a text display format. A style can have properties, which can be applied to individual paragraphs or runs. Styles make runs more compact by reducing the number of repeated definitions and properties, and the amount of work required to make changes to the document's appearance. With styles, the appearance of all the pieces of text that share a common style can be changed in one place, in that style's definition.

A series of paragraphs can have *numbering* applied to them via a numbering definition instance or a numbering style.

Data in a WordprocessingML document can be organized in a *table*, a two-dimensional grid of *cells* organized into *rows* and *columns*. Cells and whole tables can have associated properties. A cell can contain text and paragraphs, for example.

Text within a WordprocessingMLdocument can be determined dynamically via the use of *fields*. Fields consist of *field instructions* (the text that dictates the field's dynamic behavior) and the *field result* (the text resulting from the dynamic calculation of the field instructions. For example, page numbers are represented as fields. A *hyperlink* consists of two pieces of information: the hyperlink itself—the text the user clicks—and the target for the link. Potential targets include external files, e-mail addresses, web sites, and bookmarks within the document itself.

A WordprocessingML document can also contain *custom markup*, user-defined semantics applied to arbitrary document content.

A WordprocessingML document is not stored as one large body in a single part; instead, the elements that implement certain groupings of functionality are stored in separate parts. For example, all footnotes in a document are stored in one footnote part, while each section can have up to three different header parts and three different footer parts, to support headers and footers on odd-numbered pages, even-numbered pages, and the first page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

8.5 SpreadsheetML

This subclause introduces the overall form of a SpreadsheetML package, and identifies some of its main components. (See Annex L for a more detailed introduction.)

A SpreadsheetML package has a relationship of type officeDocument, which specifies the location of the main part in the package. For a SpreadsheetML document, that part contains the workbook definition.

A SpreadsheetML package's main part starts with a spreadsheet root element. That element is a *workbook*, which refers to one or more *worksheets*, which, in turn, contain the data. A worksheet is a two-dimensional grid of *cells* that are organized into *rows* and *columns*.

The cell is the primary place in which data is stored and operated on. A cell can have a number of characteristics, such as numeric, text, date, or time *formatting*; *alignment*; *font*; *color*; and a *border*. Each cell is identified by a *cell reference*, a combination of its column and row headings.

Each horizontal set of cells in a worksheet is called a *row*, and each row has a heading numbered sequentially, starting at 1. Each vertical set of cells in a worksheet is called a *column*, and each column has an alphabetic heading named sequentially from A–Z, then AA–AZ, BA–BZ, and so on.

Instead of data, a cell can contain a *formula*, which is a recipe for calculating a value. Some formulas—called *functions*—are predefined, while others are user-defined. Examples of predefined formulas are `AVERAGE`, `MAX`, `MIN`, and `SUM`. A function takes one or more arguments on which it operates, producing a result. For example, in the formula `SUM(B1:B4)`, there is one argument, `B1:B4`, which is the range of cells B1–B4, inclusive.

Other features that a SpreadsheetML document can contain include the following: *comments*, *hyperlinks*, *images*, and sorted and filtered *tables*.

A SpreadsheetML document is not stored as one large body in a single part; instead, the elements that implement certain groupings of functionality are stored in separate parts. For example, all the data for a worksheet is stored in that worksheet's part, all string literals from all worksheets are stored in a single shared string part, and each worksheet having comments has its own comments part.

8.6 PresentationML

This subclause introduces the overall form of a PresentationML package, and identifies some of its main components. (See Annex L for a more detailed introduction.)

A PresentationML package has a relationship of type officeDocument, which specifies the location of the main part in the package. For a PresentationML document, that part contains the presentation definition.

A PresentationML package's main part starts with a presentation root element. That element contains a *presentation*, which, in turn, refers to a *slide* list, a *slide master* list, a *notes master* list, and a *handout master* list. The slide list refers to all of the slides in the presentation; the slide master list refers to all of the slide masters used in the presentation; the notes master contains information about the formatting of notes pages; and the handout master describes how a handout looks.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A *handout* is a printed set of slides that can be handed out to an *audience* for future reference.

As well as text and graphics, each slide can contain *comments* and *notes*, can have a *layout*, and can be part of one or more *custom presentations*. (A comment is an annotation intended for the person maintaining the presentation slide deck. A note is a reminder or piece of text intended for the presenter or the audience.)

Other features that a PresentationML document can contain include the following: *animation*, *audio*, *video*, and *transitions* between slides.

A PresentationML document is not stored as one large body in a single part; instead, the elements that implement certain groupings of functionality are stored in separate parts. For example, all comments in a document are stored in one comment part while each slide has its own part.

8.7 Supporting MLs

This subclause introduces the set of markup languages used across package categories. (See Annex L for a more detailed introduction.)

The three markup languages described above define the structure of a package that is either a document (WordprocessingML), a spreadsheet (SpreadsheetML), or a presentation (PresentationML). However, there is also a set of shared markup languages used for common elements such as charts, diagrams, and drawing objects. These MLs are discussed below.

8.7.1 DrawingML

DrawingML specifies the location and appearance of drawing elements in a package. For example, these elements could be, but are not limited to, shapes, pictures, and tables. The root element of a DrawingML XML fragment specifies the presence of a drawing at this location in the document.

A *shape* is a geometric object such as a circle, square, or rectangle; a *picture* is an image presented inside the document; and a *table* is a two-dimensional grid of *cells* organized into *rows* and *columns*. Cells and whole tables can have associated properties. A cell can contain text, for example.

DrawingML also specifies the location and appearance of charts in a package. The root element of a chart part is chart, and specifies the appearance of the chart at this location in the document.

In addition, DrawingML specifies package-wide appearance characteristics, such as the package's theme. The *theme* of a document specifies the *color scheme*, *fonts*, and *effects*, which can be referenced by parts of the document—such as text, drawings, charts, and diagrams—in order to create a consistent visual presentation.

A *chart* is a presentation of data in a graphical fashion, such as a pie chart, bar chart, line chart, in order to make trends and exceptions in the data more visually apparent.

DrawingML also specifies the location and appearance of diagrams in a document. Together, the following four parts define a diagram:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- The *data* part (§14.2.4) specifies individual items of information presented in the diagram. Typically, each piece is a simple line of text, but depending on the diagram, an item of data might also be an image.
- The *layout* part (§14.2.5) specifies how the data and shapes are laid out to create the resulting diagram.
- The *colors* part (§14.2.3) specifies the color which is applied to each individual shape in the diagram.
- The *styles* part (§14.2.6) defines how each individual shape in the diagram maps to the document's theme.

8.7.2 Custom XML Data Properties

Custom XML Data properties allow the ability to store arbitrary XML in a package, along with schema information used by that XML.

8.7.3 File Properties

The *core file properties* of a package enable users to discover, get, and set common sets of properties from within that package, regardless of whether it's a WordprocessingML, SpreadsheetML, or PresentationML package, or another use of OPC. Such properties include creator name, creation date, title, and description.

Extended file properties are specific to Office Open XML packages. For example, for a WordprocessingML package, these properties include the number of characters, words, lines, paragraphs, and pages in the document. For a SpreadsheetML package, these properties include worksheet titles. For a PresentationML package, these properties include presentation format, the number of slides, the number of notes, and whether or not any slides are hidden.

Custom file properties are defined by the user. Examples include the name of the client for whom the document was prepared, a date/time on which some event happened, a document number, or some Boolean status flag. Each custom file property has a value, and that value has a data type.

8.7.4 Math

Math is used, mainly in documents, to specify the structure and appearance of equations. The outermost root element can be either oMath or oMathPara, the latter being a math paragraph with one or more equations where each equation is specified using a single oMath element

8.7.5 Bibliography

Bibliography specifies the structure for all references stored within a document, for use in citations or a bibliography.

End of informative text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

9. Packages

An Office Open XML document is stored as a package, whose format is defined by ISO/IE 29500-2. This subclause contains information regarding Office Open XML's use of OPC.

Throughout ISO/IEC 29500, the Open Packaging Conventions are referred to by their abbreviated name, OPC.

9.1 Office Open XML's Use of OPC

While the OPC specification is designed for the representation of Office Open XML documents, it could also support a much broader range of applications. Clarifications to the use of OPC are discussed in the following subordinate subclauses. Any requirement not mentioned here is inherited from the OPC specification.

9.1.1 Part Addressing

Parts in an Office Open XML package targeted by relationships are addressed in relationship markup through part names. External document resources targeted by a relationship can be addressed using both relative and absolute references.

9.1.2 Fragments

Fragment identifiers are supported as part of all Office Open XML external relationship targets and some Office Open XML internal relationship targets.

9.1.3 Physical Packages

Each Office Open XML document is implemented as a ZIP archive.

9.1.4 Unknown Parts

With the exception of relationship parts, all other parts in an Office Open XML document that are not the target of an implicit or explicit relationship are considered *unknown parts*. Unknown parts shall be ignored on document consumption and can, but need not, be discarded on production.

9.1.5 Trash Items

Trash items represent parts that have been discarded or are no longer in use. Trash items shall not conform to OPC part naming guidelines as defined in ISO/IEC 29500-2 and shall not be associated with a content type. All trash items shall follow the naming scheme: `[trash]/HHHH.dat` where H represents a hexadecimal digit.

[*Example*: A package has two parts that must be updated in-place but both parts have grown beyond their growth hints. The newer updated parts are added as new ZIP items while the original parts are renamed to:

```
[trash]/0000.dat
[trash]/0001.dat
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

9.1.6 Invalid Parts

ZIP archive items that do not conform to OPC part naming guidelines or are not associated with a content type shall not be allowed in an Office Open XML document, with the exception of items specifically defined by ISO/IEC 29500-2 and trash items.

9.1.7 Unknown Relationships

All relationships not defined within ISO/IEC 29500 are considered *unknown relationships*. Unknown relationships are permitted within an Office Open XML document provided that they conform to relationship markup guidelines as defined by the OPC specification. Specifically:

- Conforming consumers shall not fail to load a document containing unknown relationships.
- Conforming producers that are also consumers can, but are not required to, roundtrip and preserve unknown relationships and their target parts.

It is a requirement of ISO/IEC 29500 that dynamic extension mechanisms, such as scripting languages and macro mechanisms, shall use, for the executable parts, the correct content types, and shall not use any of the content types already defined in ISO/IEC 29500.

[*Guidance:* This subclause defines the general behavior for the consumption of unknown relationships. A conforming producer that wishes to store custom business data in an Office Open XML document should use instead the known relationship type for Custom XML Data Storage parts, as defined in §15.2.4. *end guidance*]

9.1.8 Interleaving

Interleaving as defined in ISO/IEC 29500-2 shall not be used for Office Open XML documents.

9.2 Relationships in Office Open XML

In OPC, relationships describe references from parts to other internal resources in the package or to external resources. They represent the type of connection between a source part and a target resource, and make the connection directly discoverable without looking at the part contents, so they are quick to resolve.

The same ZIP item can be the target of multiple relationships. [*Note*: Having multiple paths to a target can make access to that target more convenient. *end note*]

Office Open XML imposes constraints on relationships, described in subsequent clauses of this Part of ISO/IEC 29500. Relationships in Office Open XML are either explicit or implicit.

For an explicit relationship, a resource is referenced from a source part's XML using the Id attribute of a Relationship tag. [*Example*: A document part can have a relationship to a hyperlink only if that hyperlink's Relationship element's Id attribute value is referenced explicitly by the document part's XML. *end example*] [*Note*: Because this mechanism is used generically across multiple XML elements, explicit relationships can be extracted from an Office Open XML document without prior knowledge of tag semantics. *end note*]. Certain

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

relationships shall be *explicit*. All other relationships are *implicit.* [*Note*: The syntax for specifying an implicit relationship varies based on the content model of the XML element which is the source of the relationship. *end note*]. Relationships that are required or permitted, and restrictions on those relationships are described in §10–15 of this Part of ISO/IEC 29500.

[*Example*: Consider a WordprocessingML document that contains the following footnote sentence fragment, "… produced by Ecma[1] (http://www.ecma-international.org/).", which contains a footnote and a hyperlink to a web site. The relationship from a source to a footnote is implicit while that to a hyperlink is explicit.

The Main Document part's relationship file contains the following:

```
<Relationships …>
  <Relationship Id="rId5" Type="…/footnotes"
    Target="footnotes.xml"/>
  <Relationship Id="rId7" Type="…/hyperlink"
    Target="http://www.ecma-international.org/" TargetMode="External"/>
</Relationships>
```

All footnotes for a WordprocessingML document are contained in the same Footnotes part. Let's look at how the Main Document refers to the footnote. At the point at which the footnote reference is inserted, the following XML is present:

```
<w:r>
      <w:footnoteReference w:id="2"/>
</w:r>
```

The `w:id="2"` refers to the footnote with `id=2` in the Footnotes part, the relevant piece of which is:

```
<w:footnote w:id="2">
  …
  Ecma is an international standards development organization (SDO).
  …
</w:footnote>
```

In the case of the hyperlink, the main document part makes an explicit reference to this relationship when it refers to the hyperlink, by using the following:

```
<w:hyperlink r:id="rId7" w:history="1">
  …
</w:hyperlink>
```

The important distinction here is that there is no explicit reference to a relationship ID designating the Footnotes part. The reference to the footnote with `id=2` is "understood" to be in the Footnotes part that must always exist if there are any footnotes in the document. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: The following figure shows how the source, relationship item, and the target relate to each other for implicit and explicit relationships, respectively. The target does not have to be a file, however.

The dots correspond to attributes of relevant elements. Where one attribute refers to a piece in another part, this is indicated by arrows. Solid arrows indicate that the value of the source directly corresponds to the value at the target (for instance, `id=rId4` in the source part corresponds to `id=rId4` in the relationship item).

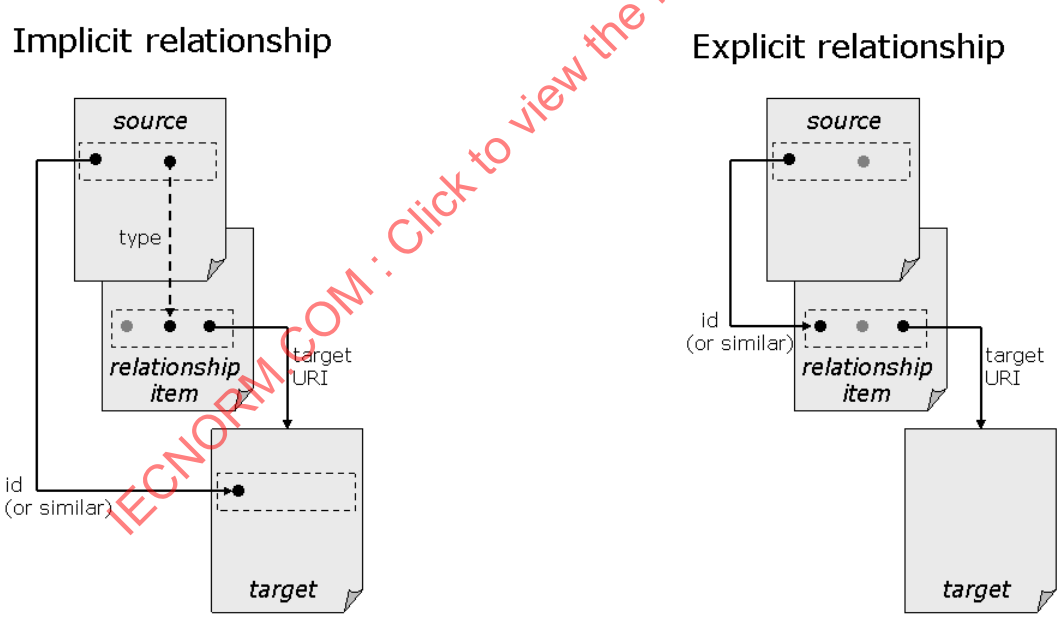
Dotted arrows indicate that the value of the source only implicit corresponds to the value of the target (for instance, "footnoteReference" in the source indicates the relationship type "footnotes" in the relationship item).

The main difference between the two types of relationship is that for implicit relationships, the id of the reference refers to an element with the same id in the target part, whereas for explicit relationships, the id refers to a relationship with the same id in the relationship item.

Both relationship types use the target URI of the relationship in the relationship item to locate the target.

For explicit relationships, the id in the source XML maps directly to the id of a relationship item with a direct explicit reference to the target. For implicit relationships, the relationship item is implied by the containing tag (e.g., footnote) and the id in the source XML is used to locate the correct element within the implied target.

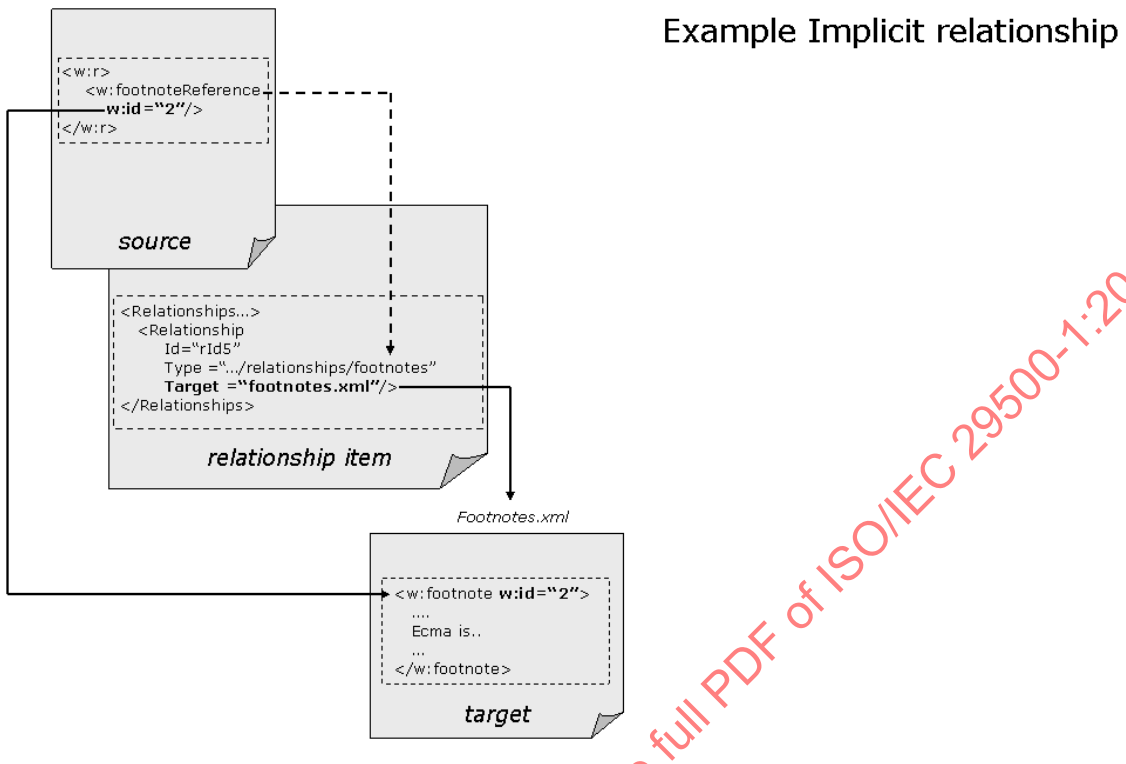
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



end example]

©ISO/IEC 2016 – All rights reserved

[*Example*: The following figure shows the implicit relationship for the footnote example described earlier.



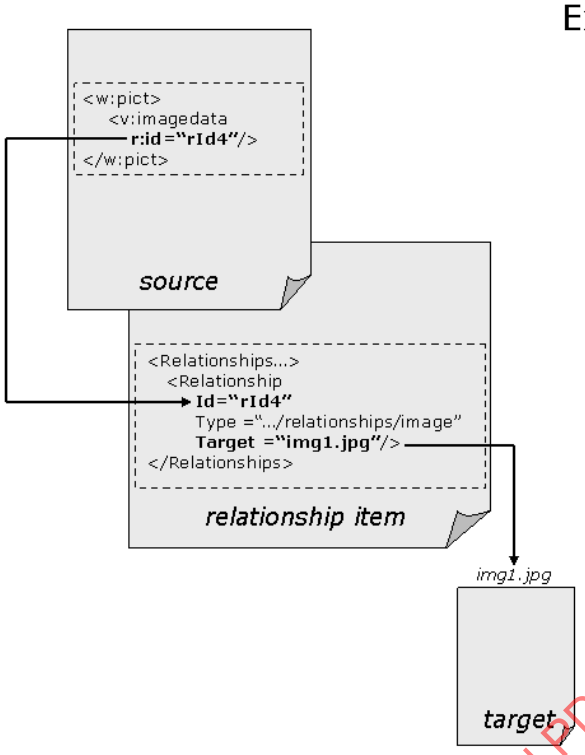
end example]

[*Example*: The following figure shows an explicit relationship.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Example Explicit relationship



end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

10. Markup Compatibility and Extensibility

Office Open XML documents are designed to allow for innovation by extending their capabilities, by using (where allowed) the Application-Defined Extension Element extLst specified by this Part of ISO/IEC 29500 or by using the Markup Compatibility and Extensibility features specified by ISO/IEC 29500-3. All the features of ISO/IEC 29500-3 are supported by this Part of ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

11. WordprocessingML

This clause contains specifications for relationship items and parts that are specific to WordprocessingML. Parts that can occur in a WordprocessingML document, but are not WordprocessingML-specific, are specified in §15.2. Unless stated explicitly, all references to relationship items, content-type items, and parts in this clause refer to WordprocessingML ZIP items.

11.1 Glossary of WordprocessingML-Specific Terms

The following terms are used in the context of a WordprocessingML document:

document setting — A document-level property that affects the handling of a given document, and influences the appearance and behavior of the current document, as well as the stored document-level state.

document building block — A reusable element in a template. [*Note*: Such elements include boilerplate text, cover pages, equations, footers, headers, tables, text boxes, and watermarks. *end note*]

glossary document — An additional WordprocessingML document story used to store reusable fragments of rich WordprocessingML content. It is called the glossary document as this story contains one or more fragments that can be indexed and extracted by name, like items in a glossary.

master document — A document that is the parent of one or more subdocuments. [*Note*: A master document can be used to manage a multipart document, such as a book having several chapters. In such as case, the master document might contain the cover page, front matter, table of contents, and cross-reference index, while each chapter and appendix resides in its own subdocument. *end note*]

section — A portion of a document in which certain page formatting options can be set. [*Note*: A new section is created to change such properties as line numbering, number of columns, or headers and footers. *end note*]

subdocument — A piece of a master document. [*Note*: A chapter or appendix might be a subdocument in a book. *end note*]

supplementary document storage location — A part within a WordprocessingML document in which fragments of WordprocessingML content can be stored separate from the printed page. See also **glossary document**

template — A document that is a pattern for creating other documents. A template can contain text, formatting, and graphics, among other things, such that documents based on it automatically have access to these elements.

11.2 Package Structure

A WordprocessingML package shall contain a package-relationship item and a content-type item. The package-relationship item shall have implicit relationships with targets of the following type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- One Main Document part (§11.3.10)

The package-relationship item is permitted to have implicit relationships with targets of the following type:

- Digital Signature Origin (§15.2.7)
- File Property parts (§15.2.12) (Application-Defined File Properties, Core File Properties, and Custom File Properties), as appropriate.
- Thumbnail (§15.2.16).

The required and optional relationships between parts are defined in §16.1 and its subordinate clauses.

[*Example*: The following package represents the minimal conformant WordprocessingML package as defined by ISO/IEC 29500:

First, the content type for relationship parts and the Main Document part (the only required part) must be defined (physically located at /[Content_Types].xml in the package):

```
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Default Extension="rels"
    ContentType="application/vnd.openxmlformats-
      package.relationships+xml"/>
  <Override PartName="/document.xml"
    ContentType="application/vnd.openxmlformats-
      officedocument.wordprocessingml.document.main+xml"/>
</Types>
```

Next, the single required relationship (the package-level relationship to the Main Document part) must be defined (physically located at /_rels/.rels in the package):

```
<Relationships xmlns="…">
  <Relationship Id="rId1"

Type="http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument"
    Target="document.xml"/>
</Relationships>
```

Finally, the minimum content for the Main Document part must be defined (physically located at /document.xml in the package):

```
<w:document xmlns:w="…">
  <w:body>
    <w:p/>
  </w:body>
</w:document>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a WordprocessingML document that is an early draft of ISO/IEC 29500. Here's an example of the hierarchical folder structure that might be used for the ZIP items in the package for that document. As shown, one part, Main Document (stored in the ZIP item /word/document.xml), has its own relationship item:

```
/[Content_Types].xml        Content-type item
/_rels/.rels                      Package-relationship item
/docProps/app.xml                 Application-Defined File Properties part
/docProps/core.xml                Core File Properties part
/word/document.xml                Main Document part
/word/_rels/document.xml.rels     Part-relationship item
/word/comments.xml                Comment part
/word/endnotes.xml                Endnotes part
/word/fontTable.xml               Font Table part
/word/footer1.xml                 Footer parts
/word/footer2.xml
/word/footer3.xml
/word/footer4.xml
/word/footnotes.xml               Footnotes part
/word/header1.xml                 Header parts
/word/header2.xml
/word/header3.xml
/word/header4.xml
/word/header5.xml
/word/header6.xml
/word/numbering.xml               Numbering Definitions part
/word/settings.xml                Document Settings part
/word/styles.xml                  Style Definitions part
/word/theme/theme1.xml            Theme part
```

The package-relationship item contains the following:

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/extended-properties" Target="docProps/app.xml"/>
  <Relationship Id="rId2"
    Type="http://…/core-properties" Target="docProps/core.xml"/>
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="word/document.xml"/>
</Relationships>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

11.3 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to WordprocessingML. [*Note*: For convenience, information from those subclauses is summarized in the following table:

Part	Relationship Target of	Root Element	Ref.
Alternative Format Import	Comments, Endnotes, Footer, Footnotes, Header, or Main Document	Not applicable	§11.3.1
Comments	Glossary Document or Main Document	comments	§11.3.2
Document Settings	Glossary Document or Main Document	settings	§11.3.3
Endnotes	Glossary Document or Main Document	endnotes	§11.3.4
Font Table	Glossary Document or Main Document	fonts	§11.3.5
Footer	Glossary Document or Main Document	ftr	§11.3.6
Footnotes	Glossary Document or Main Document	footnotes	§11.3.7
Glossary Document	Main Document	glossaryDocument	§11.3.8
Header	Glossary Document or Main Document	hdr	§11.3.9
Main Document	WordprocessingML package	document	§11.3.10
Numbering Definitions	Glossary Document or Main Document	numbering	§11.3.11
Style Definitions	Glossary Document or Main Document	styles	§11.3.12
Web Settings	Glossary Document or Main Document	webSettings	§11.3.13

end note]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

11.3.1 Alternative Format Import Part

Content Type:	Any text-based content, support for which is application-defined. [*Note*: Some examples of formats which might be supported include: • Text = text/plain • HTML = text/html • WordprocessingML = application/vnd.openxmlformats-officedocument.wordprocessingml.document • XHTML = application/xhtml+xml

©ISO/IEC 2016 – All rights reserved

	end note]
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/aFChunk

An alternative format import part allows content specified in an alternate format specified above to be embedded directly in a WordprocessingML document in order to allow that content to be migrated to the WordprocessingML format.

Any document part that permits a p element can also contain an altChunk element, whose id attribute refers to a relationship. That relationship shall target a part within the package, which contains the content to be imported into this WordprocessingML document.

A package is permitted to contain zero or more Alternative Format Import parts, each of which shall have a corresponding alternate format file that is the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part. This relationship shall be explicitly referenced using its relationship ID in the source part using the appropriate XML syntax (i.e.; in the id attribute on the altChunk element), and the presence of this relationship without such a reference shall be considered non-conformant.

ISO/IEC 29500 does not specify how one might create a WordprocessingML package that contains Alternative Format Import relationships and altChunk elements.

The following requirements are applied to applications with respect to this part:

- An application that is solely a conforming consumer shall not reject documents containing one or more instances of this part
- An application that is both a conforming consumer and producer shall not reject documents containing instances of this part and shall convert/remove any instances of this part before acting as producer.
- An application that is solely a conforming producer shall not create a WordprocessingML package that contains Alternative Format Import relationships and elements.

[*Note*: The Alternative Format Import machinery provides a one-time conversion facility. A producer could have an extension that allows it to generate a package containing these relationships and elements, yet when run in conforming mode, does not do so. *end note*]

[*Example*: The following Main Document part-relationship item contains a relationship to an Alternative Format Import part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/aFChunk" Target="Demo.html"
    TargetMode="Internal"/>
</Relationships>
```

The Main Document part contains the following XML fragment:

```
<w:body>
  …
  <w:p/>
  <w:altChunk r:id="rId5"/>
  <w:p/>
  …
</w:body>
```

which results in the entire contents of Demo.html being converted and brought into the document at that point (assuming that the content type of Demo.html is supported by the application consuming this WordprocessingML file). *end example*]

An Alternative Format Import part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

An Alternative Format Import part shall not have any explicit or implicit relationships to parts defined by ISO/IEC 29500.

A producer that wants interoperability should use one of the following standard formats:

- HTML - application/text/html
- TEXT - application/text/plain (UTF-16)

11.3.2 Comments Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.comments+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/comments

An instance of this part type contains the information about each comment in the document.

A package shall contain no more than two Comments parts. If it exists, one instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following Main Document part-relationship item contains a relationship to the Contents part, which is stored as the ZIP item comment.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId93"
    Type="http://…/comments" Target="comments.xml"/>
</Relationships>
```

end example]

The root element for a Comment part shall be comments.

[*Example*:

```
<w:comments … >
  <w:comment>
    …
  </w:comment>
  …
</w:comments>
```

end example]

The XML markup for a comment in a Main Document part uses the commentReference element.

[*Example*: Consider the case in which the Main Document part contains the text "… in the Standard.", and there is an comment inserted immediately after the period:

```
<w:p …>
  …
  <w:r>
    <w:t>… in the Standard.</w:t>
  </w:r>
  <w:r>
    <w:commentReference w:id="1"/>
  </w:r>
</w:p>
```

end example]

Each comment has a corresponding comment element in the Comments part, which contains the text of the comment.

[*Example*: The text of the comment is "This is my comment.":

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:comments xmlns:w="…"
  <w:comment w:id="1">
    <w:p>
      <w:r>
        <w:t>This is my comment.</w:t>
    </w:p>
  </w:comment>
</w:comments>
```

end example]

A Comments part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Comments part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Video (§15.2.17)

A Comments part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.3 Document Settings Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.settings+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/settings

An instance of this part type contains all the document's properties.

A package shall contain no more than two Document Settings parts. If it exists, one instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: The following Main Document part-relationship item contains a relationship to a Document Settings part, which is stored in the ZIP item settings.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/settings" Target="settings.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be settings.

[*Example*:

```
<w:settings … >
  …
  <w:defaultTabStop w:val="360"/>
  <w:footnotePr>
    …
  </w:footnotePr>
  <w:endnotePr>
  …
  </w:endnotePr>
  <w:rsids>
  …
  </w:rsids>
  …
</w:settings>
```

end example]

A Document Settings part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Document Settings part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Document Template (§11.4)
- Mail Merge Data Source (§11.7)
- Mail Merge Header Data Source (§11.8)
- XSL Transformation (§11.9)

A Document Settings part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

11.3.4 Endnotes Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.endnotes+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/endnotes

An instance of this part type contains all the endnotes for the document.

A package shall contain no more than two Endnotes parts. If it exists, one instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

[*Example*: The following Main Document part-relationship item contains a relationship to the Endnotes part, which is stored as the ZIP item endnotes.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId6"
    Type="http://…/endnotes" Target="endnotes.xml"/>
</Relationships>
```

end example]

The root element for an Endnotes part shall be endnotes.

[*Example*:

```
<w:endnotes xmlns:w="…" …>
  <w:endnote …>
    …
  </w:endnote>
  <w:endnote …>
    …
  </w:endnote>
</w:endnotes>
```

end example]

The XML markup for an endnote in a Main Document part uses the endnoteReference element.

[*Example*: Consider the case in which the Main Document part contains the text "… in the Standard.", and there is an endnote inserted immediately after the period:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p …>
  …
  <w:r>
    <w:t>… in the Standard.</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="EndnoteReference"/>
    </w:rPr>
    <w:endnoteReference w:id="5"/>
  </w:r>
</w:p>
```

end example]

Each endnote has a corresponding endnote element in the Endnotes part, which contains the text of the endnote, and the endnoteRef element.

[*Example*: The text of the endnote is "This can be downloaded from http://www.aabbcc.com/index.html." where "http://www.aabbcc.com/index.html" is marked as a hyperlink:

```
<w:endnotes xmlns:w="…">
  <w:endnote w:id="5">
    <w:p>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="EndnoteReference"/>
        </w:rPr>
        <w:endnoteRef/>
      </w:r>
      <w:r>
        <w:t xml:space="preserve"> This can be downloaded from </w:t>
      </w:r>
      <w:hyperlink r:id="rId2">
        <w:r>
          <w:rPr>
            <w:rStyle w:val="Hyperlink"/>
          </w:rPr>
          <w:t>http://www.aabbcc.com/index.html</w:t>
        </w:r>
      </w:hyperlink>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <w:r>
          <w:t>.</w:t>
        </w:p>
      </w:endnote>
    </w:endnotes>
```

end example]

An Endnotes part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

An Endnotes part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Video (§15.2.17)

An Endnotes part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.5 Font Table Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.fontTable+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/fontTable

An instance of this part type contains information about each of the fonts used by content in the document. When a consumer reads a WordprocessingML document, it shall use this information to determine which fonts to use to display the document when the specified fonts are not available on the consumer's system.

A package shall contain no more than two Font Table parts. If it exists, one instance of that part shall be the target of an implicit relationship in the part-relationship item for the Main Document (§11.3.10) part, and the other instance shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following Main Document part-relationship item contains a relationship to the Font Table part, which is stored as the ZIP item fontTable.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/fontTable" Target="fontTable.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be fonts.

[*Example*:

```
<w:fonts … >
  <w:font w:name="Calibri">
    <w:panose1 w:val="020F0502020204030204"/>
    <w:charset w:val="00"/>
    <w:family w:val="swiss"/>
    <w:pitch w:val="variable"/>
    <w:sig w:usb0="A00002EF" w:usb1="4000207B" w:usb2="00000000"
      w:usb3="00000000" w:csb0="0000009F" w:csb1="00000000"/>
  </w:font>
</w:fonts>
```

end example]

A Font Table part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Font Table part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Fonts (§15.2.13)

A Font Table part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.6 Footer Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.footer+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/footer

An instance of this part type contains the information about a footer displayed for one or more sections.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package is permitted to contain zero or one Footer part for each kind of footer (first page, odd page, or even page) in each section of the document. Each Footer part shall be the target of an explicit relationship in the part-relationship item for the Main Document (§11.3.10) part, or the Glossary Document (§11.3.8) part.

[*Example*: The Main Document part-relationship item contains one relationship, for the odd footer part, which is stored as the ZIP item footer3.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId91"
    Type="http://…/footer" Target="footer3.xml"/>
</Relationships>
```

end example]

The root element for a Footer part shall be ftr.

[*Example*:

```
<w:ftr xmlns:w="…" …>
  …
</w:ftr>
```

end example]

The XML markup for a footer in a section of a Main Document part involves the footerReference element in that section's sectPr element which explicitly references the relationship for the header.

[*Example*: Consider the case in which a section in the Main Document part contains odd and even headers, and an odd footer:

```
<w:document xmlns:w="…">
  …
  <w:sectPr>
    <w:footerReference w:val="rId89" w:type="default"/>
    <w:footerReference w:val="rId90" w:type="even"/>
    <w:footerReference w:val="rId91" w:type="first"/>
    <w:type w:val="oddPage"/>
    <w:pgSz w:w="11909" w:h="16834" w:code="9"/>
    <w:pgMar w:top="1440" w:right="1152" w:bottom="1440"
      w:left="1152" w:header="720" w:footer="720" w:gutter="0"/>
    <w:lnNumType w:countBy="1"/>
    <w:pgNumType w:numFmt="lowerRoman"/>
    <w:cols w:space="720"/>
  </w:sectPr>
</w:document>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

Each footer has a corresponding ftr element in a Footer part, which contains the text of the footer.

[*Example*: Here is the odd footer corresponding to the example above. It has the page number centered and displayed using lowercase Roman numerals (as set by the pgNumType element above):

```
<w:ftr xmlns:w="…">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="Centered"/>
    </w:pPr>
    <w:fldSimple w:instr="PAGE">
      <w:r>
        <w:rPr>
          <w:noProof/>
        </w:rPr>
        <w:t>i</w:t>
      </w:r>
    </w:fldSimple>
  </w:p>
</w:ftr>
```

end example]

A Footer part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Footer part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Video (§15.2.17)

A Footer part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

11.3.7 Footnotes Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.footnotes+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/footnotes

An instance of this part type contains all the footnotes for the document.

A package shall contain no more than two Footnotes parts. If it exists, one instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

[*Example*: The Main Document part-relationship item contains a relationship to the Footnotes part, which is stored as the ZIP item footnotes.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/footnotes" Target="footnotes.xml"/>
</Relationships>
```

end example]

The root element for a Footnotes part shall be footnotes.

[*Example*:

```
<w:footnotes xmlns:w="…" …>
  <w:footnote …>
    …
  </w:footnote>
  <w:footnote …>
    …
  </w:footnote>
</w:footnotes>
```

end example]

The XML markup for a footnote in a Main Document part involves the footnoteReference element.

[*Example*: Consider the case in which the Main Document part contains the text "… in the Standard.", and there is a footnote inserted immediately after the period:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p …>
  …
  <w:r>
    <w:t>… in the Standard.</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="FootnoteReference"/>
    </w:rPr>
    <w:footnoteReference w:id="5"/>
  </w:r>
</w:p>
```

end example]

Each footnote has a corresponding footnote element in the Footnotes part, which contains the text of the footnote and the footnoteRef element.

[*Example*: The text of the footnote is "This can be downloaded from http://www.aabbcc.com/index.html." where "http://www.aabbcc.com/index.html" is marked as a hyperlink:

```
<w:footnotes xmlns:w="…"
  <w:footnote w:id="5">
    <w:p>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="FootnoteReference"/>
        </w:rPr>
        <w:footnoteRef/>
      </w:r>
      <w:r>
        <w:t xml:space="preserve">This can be downloaded from </w:t>
      </w:r>
      <w:hyperlink r:id="rId2" w:history="1">
        <w:r>
          <w:rPr>
            <w:rStyle w:val="Hyperlink"/>
          </w:rPr>
          <w:t>http://www.aabbcc.com/index.html</w:t>
        </w:r>
      </w:hyperlink>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <w:r>
          <w:t>.</w:t>
        </w:r>
      </w:p>
    </w:footnote>
  </w:footnotes>
```

end example]

A Footnotes part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Footnotes part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Video (§15.2.17)

A Footer part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.8 Glossary Document Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.document.glossary+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/glossaryDocument

An instance of this part type is a supplementary document storage location which stores the definition and content for content that shall be carried with the document for future insertion and/or use, but which shall not be visible within the contents of the main document story. [*Example*: A legal contract template might include one or more optional clauses that shall not appear in the document until those clauses are inserted explicitly via a user action. To store these optional clauses until they are inserted, their contents are placed in the glossary document part. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: This part is intended for storage of optional "document fragments" which are often used to perform document assembly. The use of the word *glossary* is a reference to the fact that each of these entries was historically referenced by its first word in legacy word processing applications, like definitions of terms in a traditional glossary. *end note*]

The root element for a part of this content type shall be glossaryDocument.

[*Example*: The following part contains two building blocks. The first block is named "rainbow colors", belongs to a category called "Misc", belongs to a gallery called "docParts", and contains the text "The colors … and violet." The details of the second block have been omitted:

```
<w:glossaryDocument xmlns:w="…" >
  <w:docParts>
    <w:docPart>
      <w:docPartPr>
        <w:name w:val="rainbow colors"/>
        <w:style w:val="Normal"/>
        <w:category>
          <w:name w:val="Misc"/>
          <w:gallery w:val="docParts"/>
        </w:category>
      </w:docPartPr>
      <w:docPartBody>
        <w:p>
          <w:r>
            <w:t>The colors of the rainbow are red, orange, yellow,
              green, blue, indigo, and violet.</w:t>
          </w:r>
        </w:p>
      </w:docPartBody>
    </w:docPart>
    <w:docPart>
      …
    </w:docPart>
  </w:docParts>
</w:glossaryDocument>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

A package shall contain at most one Glossary Document part, and that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part.

[*Example*: The following Main Document part-relationship item contains a relationship to a Glossary Document part, which is stored in the ZIP item `glossary/document.xml`:

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/glossaryDocument" Target="glossary/document.xml"/>
</Relationships>
```

end example]

A Glossary Document part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Glossary Document part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Comments (§11.3.2)
- Document Settings (§11.3.3)
- Endnotes (§11.3.4)
- Font Table (§11.3.5)
- Footnotes (§11.3.7)
- Numbering Definitions (§11.3.11)
- Style Definitions (§11.3.11)

A Glossary Document part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors (§14.2.3), Diagram Data (§14.2.4), Diagram Layout Definition (§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Footer (§11.3.6)
- Header (§11.3.9)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Printer Settings (§15.2.15)
- Video (§15.2.17)

A Glossary Document part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

11.3.9 Header Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/header

An instance of this part type contains the information about a header displayed for one or more sections.

A package shall contain zero or one Header part for each kind of header (first page, odd page, or even page) in each section of the document. Each Header part shall be the target of an explicit relationship from the Main Document (§11.3.10) part or the Glossary Document (§11.3.8) part.

[*Example*: The Main Document part-relationship item contains two relationships: one for the even header part (which is stored as the ZIP item header2.xml) and one for the odd header part (which is stored as the ZIP item header3.xml):

```
<Relationships xmlns="…">
  <Relationship Id="rId89" Type="http://…/header" Target="header2.xml"/>
  <Relationship Id="rId90" Type="http://…/header" Target="header3.xml"/>
</Relationships>
```

end example]

The root element for a Header part shall be hdr.

[*Example*:

```
<w:hdr xmlns:w="…" …>
  …
</w:hdr>
```

end example]

The XML markup for a header in a section of a Main Document part involves the headerReference element in that section's sectPr element.

[*Example*: Consider the case in which a section in the Main Document part contains odd and even headers, and an odd footer:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:body>
  …
  <w:sectPr w:rsidR="00363F31" w:rsidSect="008D4B40">
    <w:headerReference w:val="rId89" w:type="default"/>
    <w:headerReference w:val="rId90" w:type="even"/>
    <w:headerReference w:val="rId91" w:type="first"/>
    <w:type w:val="oddPage"/>
    <w:pgSz w:w="11909" w:h="16834" w:code="9"/>
    <w:pgMar w:top="1440" w:right="1152" w:bottom="1440"
      w:left="1152" w:header="720" w:footer="720" w:gutter="0"/>
    <w:lnNumType w:countBy="1"/>
    <w:pgNumType w:fmt="lowerRoman"/>
    <w:cols w:space="720"/>
  </w:sectPr>
</w:body>
```

end example]

Each header has a corresponding hdr element in a Header part, which contains the text of the header.

[*Example*: Here is the even header corresponding to the examples above:

```
<w:hdr xmlns:w="…">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="Header"/>
    </w:pPr>
    <w:r>
      <w:t>My Test Document</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

Here is the odd header corresponding to the examples above:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:hdr xmlns:w="…">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="Header"/>
    </w:pPr>
    <w:r>
      <w:tab/>
      <w:t>Table of Contents</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

end example]

A Header part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Header part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3).
- Images (§15.2.14)
- Video (§15.2.17)

A Header part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

11.3.10 Main Document Part

Content Type(s):	application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml application/vnd.openxmlformats-officedocument.wordprocessingml.template.main+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument

An instance of this part type contains the body of the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package shall contain a Main Document part (§11.3.10) part. The Main Document part shall be the target of a relationship in the package-relationship item.

The root element for a part of this content type shall be document.

[*Example*: Given the following package-relationship item excerpt:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="word/document.xml"/>
</Relationships>
```

/word/document.xml" contains the following:

```
<w:document …>
  <w:body>
    …
  </w:body>
</w:document>
```

end example]

A Main Document part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Main Document part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Comments (§11.3.2)
- Custom XML Data Storage (§15.2.4)
- Document Settings (§11.3.3)
- Endnotes (§11.3.4)
- Font Table (§11.3.5)
- Footnotes (§11.3.7)
- Glossary Document (§11.3.8)
- Numbering Definitions (§11.3.11)
- Style Definitions (§11.3.12)
- Theme (§14.2.7)
- Thumbnail (§15.2.16)

A Main Document part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Footer (§11.3.6)
- Header (§11.3.9)
- Hyperlinks (§15.3)
- Images (§15.2.14)
- Printer Settings (§15.2.15)
- Subdocument (§11.6)
- Video (§15.2.17)

A Main Document shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.11 Numbering Definitions Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.numbering+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/numbering

An instance of this part type contains a definition for the structure of each unique numbering definition in this document.

[*Example:* If a set of paragraphs are added to a document which have a circle bullet at the first level, a square bullet at the second level, and a checkmark bullet at the third level, such as the following:

- First level
  - Second level
    - ✓ Third level

The numbering definition part contains the definition for each of these levels (their bullet style, indent, etc.) even if the second and third levels are not actually used in the document *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package shall contain no more than two Numbering Definitions parts. If they exist, one instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

[*Example*:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/numbering" Target="numbering.xml"/>
</Relationships>
```

end example]

The XML markup for a list usage involves a reference to a numbering definition via the child elements of the numPr element.

[*Example*: Here we have a paragraph set using the style `Text`, followed by a list of things which have the paragraph style `ListBullet`, followed by another paragraph set using the style `Text`:

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="Text"/>
  </w:pPr>
  <w:r>
    <w:t>The kinds of fruit needed are:</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pStyle w:val="ListBullet"/>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Apples</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="ListBullet"/>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Oranges</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pStyle w:val="Text"/>
  </w:pPr>
  <w:r>
    <w:t>Other items may be needed too.</w:t>
  </w:r>
</w:p>
```

end example]

The root element for a Numbering Definition part shall be numbering, with each numbering definition being defined by an abstractNum element.

[*Example*:

```
<w:numbering xmlns:w="…">
  <w:abstractNum w:numId="11">
    <w:nsid w:val="394E2425"/>
    <w:multiLevelType w:val="hybridMultilevel"/>
    <w:tmpl w:val="F628E89A"/>
    <w:lvl w:ilvl="0" w:tplc="151C4798">
      <w:start w:val="1"/>
      <w:numFmt w:val="bullet"/>
      <w:pStyle w:val="ListBullet"/>
      <w:lvlText w:val="…"/>
      <w:lvlJc w:val="start"/>
      <w:pPr>
        <w:tabs>
          <w:tab w:val="num" w:pos="720"/>
        </w:tabs>
        <w:ind w:start="720" w:hanging="360"/>
      </w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <w:rPr>
          <w:rFonts w:ascii="Symbol" w:hAnsi="Symbol" w:hint="default"/>
        </w:rPr>
      </w:lvl>
      …
    </w:abstractNum>
  </w:numbering>
```

end example]

A Numbering Definitions part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Numbering Definitions part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Images (§15.2.14)

A Numbering Definitions part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

11.3.12 Style Definitions Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.styles+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/styles

An instance of this part type contains the definition for a set of styles used by this document.

A package shall contain at most two Style Definitions parts. One instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship in from the Glossary Document (§11.3.8) part.

[*Example*:

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/styles" Target="styles.xml"/>
</Relationships>
```

end example]

The root element for a Styles Definition part shall be styles, which is a container for one or more style elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Here is the style `ListBullet` (which is used in a Main Document Part in §11.3.10):

```
<w:styles xmlns:w="…" … xml:space="preserve">
  <w:style w:type="paragraph" w:styleId="ListBullet">
    <w:name w:val="List Bullet"/>
    <w:basedOn w:val="Text"/>
    <w:autoRedefine/>
    <w:rsid w:val="00081289"/>
    <w:pPr>
      <w:pStyle w:val="ListBullet"/>
      <w:numPr>
        <w:numId w:val="1"/>
      </w:numPr>
      <w:tabs>
        <w:tab w:val="clear" w:pos="360"/>
      </w:tabs>
      <w:ind w:start="648"/>
    </w:pPr>
  </w:style>
</w:styles>
```

end example]

A Style Definitions part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Style Definitions part shall not have implicit or explicit relationships to any part defined by ISO/IEC 29500.

11.3.13 Web Settings Part

Content Type:	application/vnd.openxmlformats-officedocument.wordprocessingml.webSettings+xml
Root Namespace:	http://purl.oclc.org/ooxml/wordprocessingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/webSettings

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An instance of this part type contains the definition for web-specific settings used by this document.

A package shall contain at most two Web Settings parts. One instance of that part shall be the target of an implicit relationship from the Main Document (§11.3.10) part, and the other shall be the target of an implicit relationship from the Glossary Document (§11.3.8) part.

[*Example*:

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/webSettings" Target="webSettings.xml"/>
</Relationships>
```

end example]

The root element for a Web Settings part shall be webSettings.

[*Example*:

```
<w:webSettings …>
  <w:frameset>
    …
    <w:frame>
      <w:sz w:val="216" />
      <w:name w:val="Frame2" />
      <w:sourceFileName r:id="rId1" />
    </w:frame>
    <w:frame>
      <w:name w:val="Frame1" />
      <w:sourceFileName r:id="rId2" />
    </w:frame>
  </w:frameset>
</w:webSettings>
```

end example]

A Web Settings part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Web Settings part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Frameset (§11.5)

A Web Settings part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

11.4 Document Template

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/attachedTemplate

A *document template* can be represented by an instance of a WordprocessingML package, and contains styles, numbering definitions, and so on that are made available when documents based on that template are edited. A WordprocessingML document can refer to another document as its document template, by having a Document

Settings part (§11.3.3) that contains an explicit relationship to the file location of the necessary document template using the id attribute on the attachedTemplate element.

[*Example*: Consider a document specifying a document template located at `c:\template.docx`:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/attachedTemplate" Target="file:///c:\template.docx"
    TargetMode="External"/>
</Relationships>
```

The document's Document Settings part contains an attachedTemplate element that explicitly references this relationship:

```
<w:settings … >
  <w:attachedTemplate r:id="rId1"/>
</w:settings>
```

end example]

11.5 Framesets

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/frame

A *frameset* is a WordprocessingML document which specifies the location and placement of other WordprocessingML documents (which, when used in this context, are referred to as *frames*). A frameset shall be represented by an instance of a WordprocessingML document with a Web Settings part (§11.3.13) whose relationship item targets each of that frameset's frames.

[*Example*: Consider a frameset document having two frames. The frameset's Web Settings part-relationships item contains the following, in which frame1.docx and frame2.docx are packages containing the corresponding frames:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/frame" Target="frame1.docx" TargetMode="External"/>
  <Relationship Id="rId2"
    Type="http://…/frame" Target="frame2.docx" TargetMode="External"/>
</Relationships>
```

The frameset document's Web Settings part contains a frameset element that references its frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:webSettings …>
  <w:frameset>
    …
    <w:frame>
      <w:sz w:val="216" />
      <w:name w:val="Frame2" />
      <w:sourceFileName r:id="rId1" />
    </w:frame>
    <w:frame>
      <w:name w:val="Frame1" />
      <w:sourceFileName r:id="rId2" />
    </w:frame>
  </w:frameset>
</w:webSettings>
```

end example]

A frame shall be represented by an instance of a WordprocessingML package.

A frame shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

11.6 Master Documents and Subdocuments

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/subDocument

A master document shall be represented by an instance of a WordprocessingML document whose Main Document (§11.3.10) part targets each of that master document's subdocuments.

[*Rationale*: Sometimes, it is convenient to deal with a document as a collection of pieces, especially when those pieces might be edited by different authors in a collaborative group. Perhaps it simply makes sense to think about a book as a collection of chapters rather than as one big document. The breaking-up of a document into such pieces can be achieved by having a master document with one or more subdocuments. *end rationale*]

[*Example*: Consider a master document, whose three subdocuments are called Start, Middle, and End, respectively. Master's Main Document part has a corresponding relationships part that contains the following, in which Start.docx, Middle.docx, and End.docx are packages containing the corresponding subdocuments:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/subDocument"
    Target="Start.docx" TargetMode="External"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <Relationship Id="rId6"
      Type="http://…/SubDocument"
      Target="Middle.docx" TargetMode="External"/>
    <Relationship Id="rId7"
      Type="http://…/SubDocument"
      Target="End.docx" TargetMode="External"/>
  </Relationships>
```

The master document's Main Document part contains subDoc elements that reference its subdocuments:

```
  <w:document xmlns:r="…" xmlns:w="…" …>
    <w:body>
      <w:p …>
        <w:pPr>
          …
        </w:pPr>
      </w:p>
      <w:subDoc r:id="rId5"/>
      …
      <w:subDoc r:id="rId6"/>
      …
      <w:subDoc r:id="rId7"/>
      …
    </w:body>
  </w:document>
```

end example]

A subdocument shall be represented by an instance of a WordprocessingML package.

A subdocument shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

11.7 Mail Merge Data Source

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeSource

A document that stores information about a mail merge operation is permitted to contain a Document Settings part (§11.3.3) whose relationship item targets the file location of the necessary data source using this relationship.

[*Example*: Consider a document specifying a mail merge whose data source is located at `http://www.openxmlformats.org/data.txt`:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/mailMergeSource"
    Target="http://www.openxmlformats.org/data.txt"
    TargetMode="External"/>
</Relationships>
```

The document's Document Settings part contains a dataSource element that explicitly references this relationship:

```
<w:settings …>
  <w:mailMerge>
    …
    <w:dataSource r:id="rId1" />
    …
  </w:mailMerge>
</w:settings>
```

end example]

A mail merge data source shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

11.8 Mail Merge Header Data Source

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeHeaderSource

A document that stores information about a mail merge operation is permitted to contain a Document Settings part (§11.3.3) whose relationship item targets the file location of the necessary header data source using this relationship.

[*Example*: Consider a document specifying a mail merge whose header data source is located at `http://www.openxmlformats.org/header.txt`:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/mailMergeHeaderSource"
    Target="http://www.openxmlformats.org/header.txt"
    TargetMode="External"/>
</Relationships>
```

The document's Document Settings part contains a headerSource element that explicitly references this relationship:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:settings …>
  <w:mailMerge>
    …
    <w:headerSource r:id="rId2" />
    …
  </w:mailMerge>
</w:settings>
```

end example]

A mail merge header data source shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

11.9 XSL Transformation

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/transform

A document can store information about an XSL Transformation which might be applied when the document is output as a single file (e.g. as XML or HTML).That information is stored in a Document Settings part (§11.3.3) whose part relationship item contains an explicit relationship to the file location of the XSL Transformation using this relationship. [*Note*: A full description of how this relationship is used (in conjunction with the saveThroughXslt element) is provided in §**17.15.1.76**. *end note*]

[*Example*: Consider a document specifying an XSL Transformation located at `http://www.openxmlformats.org/test.xsl`:

```
<Relationships xmlns="…">
  <Relationship Id="rId8" Type="http://…/transform"
    Target="http://www.openxmlformats.org/test.xsl"
    TargetMode="External"/>
</Relationships>
```

The document's Document Settings part contains a saveThroughXslt element that explicitly references this relationship:

```
<w:settings …>
  …
  <w:saveThroughXslt r:id="rId8" />
  …
</w:settings>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

An XSL transformation shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

12. SpreadsheetML

This clause contains specifications for relationship items and parts that are specific to SpreadsheetML. Parts than can occur in a SpreadsheetML document, but are not SpreadsheetML-specific, are specified in §15.2. Unless stated explicitly, all references to relationship items, content-type items, and parts in this clause refer to SpreadsheetML ZIP items.

12.1 Glossary of SpreadsheetML-Specific Terms

The following terms are used in the context of a SpreadsheetML document:

AutoFilter – A SpreadsheetML document state in which only certain rows are displayed, determined via filter criteria applied to the columns.

cell — The location at the intersection of a row and column, in which numeric or textual data or a formula is stored. A cell can have a number of characteristics, such as numeric or text formatting, alignment, font, color, and border.

cell reference — An individual cell's designation using a combination of its column and row headings, as in A13, H19, and BX1200. A **relative cell reference** in a formula automatically changes when the formula is copied down a column or across a row. An **absolute cell reference** is fixed. Absolute references don't change when a formula is copied from one cell to another. A **mixed cell reference** has either an absolute column and a relative row, or an absolute row and a relative column.

chart — A graphical representation of data, as in a bar, column, line, pie chart, for example.

column — Any vertical set of cells in a worksheet. Each column has an alphabetic heading. Columns are named sequentially, going from A–Z, then AA–AZ, BA–BZ, and so on.

connection — The means by which external data—that is, data stored outside of a workbook (in a database or on a Web server, for example)—can be imported into a worksheet.

formula — A recipe for calculating a value. Some formulas are predefined; others are user-defined.

function — A predefined formula, such as AVERAGE, MAX, MIN, and SUM. A function takes one or more arguments on which it operates, producing a result. [*Note*: In the formula =SUM(B1:B4), there is one argument, B1:B4, which is the range of cells B1–B4, inclusive. *end note*]

MDX — A multi-dimensional expression language, passed to an OLAP provider. The method of interpreting of this is defined by the server-side OLAP provider implementation.

OLAP — A type of online analytical processing database which uses a multi-dimensional data model.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

pivot table — A kind of table that is used to manage and analyze related data that is stored elsewhere.

row — Any horizontal set of cells in a worksheet. Each row has a numeric heading. Rows are numbered sequentially, starting at 1.

table — A rectangular-shaped set of related rows and columns that can be sorted, filtered, and totaled as a group. Rows in a table can be hidden by applying **autofilters** to one or more columns.

workbook — A collection of worksheets.

worksheet — A two-dimensional grid of cells that are organized into rows and columns.

12.2 Package Structure

A SpreadsheetML package shall contain a package-relationship item and a content-type item. The package-relationship item shall have implicit relationships with targets of the following type:

- One Workbook part (12.3.23).

The package-relationship item is permitted to have implicit relationships with targets of the following type:

- Digital Signature Origin (§15.2.7)
- File Property parts (§15.2.12) (Application-Defined File Properties, Core File Properties, and Custom File Properties), as appropriate.
- Thumbnail (§15.2.16).

The required and optional relationships between parts are defined in §12.3 and its subordinate clauses.

[*Example*: The following package represents the minimal conformant SpreadsheetML package as defined by ISO/IEC 29500:

First, the content types for relationship parts, the Workbook part, and at least one Sheet part must be defined (physically located at /[Content_Types].xml in the package):

```
<Types xmlns="…">
  <Default Extension="rels"
   ContentType="application/vnd.openxmlformats-package.relationships+xml"
   />
  <Override PartName="/workbook.xml"
   ContentType="application/vnd.openxmlformats-officedocument.
   spreadsheetml.sheet.main+xml" />
  <Override PartName="/sheet1.xml"
   ContentType="application/vnd.openxmlformats-
   officedocument.spreadsheetml.worksheet+xml" />
</Types>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Next, the required package-level relationship to the Workbook part must be defined (physically located at /_rels/.rels in the package):

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
   Type=http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument"
   Target="workbook.xml" />
</Relationships>
```

Next, the minimum content for the Workbook part must be defined (physically located at /workbook.xml in the package):

```
<workbook xmlns="…" xmlns:r="…">
  <sheets>
    <sheet name="1" sheetId="1" r:id="rId1" />
  </sheets>
</workbook>
```

Next, the required workbook-level relationship to the single Sheet part must be defined, (physically located at /_rels/workbook.xml.rels in the package):

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
   Type="http://purl.oclc.org/ooxml/officeDocument/relationships/worksheet"
Target="sheet1.xml" />
</Relationships>
```

Finally, the minimum content for a single Sheet part must be defined (physically located at /sheet1.xml in the package):

```
<worksheet xmlns="…" xmlns:r="…">
  <sheetData />
</worksheet>
```

end example]

[*Example*: Consider a SpreadsheetML document that contains a workbook having three worksheets. Here's an example of the hierarchical folder structure that might be used for the ZIP items in the package for that document. As shown, one part, Workbook (stored in the ZIP item /xl/workbook.xml), has its own relationship item:

```
/_rels/.rels                    Package-relationship item
/[Content_Types].xml            Content-type item
/docProps/app.xml               Application-Defined File Properties part
/docProps/core.xml              Core File Properties part
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
/xl/workbook.xml                    Workbook part
/xl/_rels/workbook.xml.rels         Part-relationship item
/xl/calcChain.xml                   Calculation Chain part
/xl/sharedStrings.xml               Shared String Table part
/xl/styles.xml                      Styles part
/xl/volatileDependencies.xml        Volatile Dependencies part
/xl/theme/theme1.xml                Theme part
/xl/worksheets/sheet1.xml           Worksheet parts
/xl/worksheets/sheet2.xml
/xl/worksheets/sheet3.xml
```

The package-relationship item contains the following:

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/extended-properties" Target="docProps/app.xml"/>
  <Relationship Id="rId2"
    Type="http://…/core-properties" Target="docProps/core.xml"/>
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="xl/workbook.xml"/>
</Relationships>
```

end example]

12.3 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to SpreadsheetML. [*Note*: For convenience, information from those subclauses is summarized in the following table:

Part	Relationship Target of	Root Element	Ref.
Calculation Chain	Workbook	calcChain	§12.3.1
Chartsheet	Workbook	chartsheet	§12.3.2
Comments	Dialogsheet, Worksheet	comments	§12.3.3
Connections	Workbook	connections	§12.3.4
Custom Property	Workbook	Not applicable	§12.3.5
Custom XML Mappings	Workbook	MapInfo	§12.3.6
Dialogsheet	Workbook	dialogsheet	§12.3.7
Drawings	Chartsheet, Worksheet	wsDr	§12.3.8
External Workbook References	Workbook	externalLink	§12.3.9
Metadata	Workbook	metadata	§12.3.10
Pivot Table	Worksheet	pivotTableDefinition	§12.3.11

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Relationship Target of	Root Element	Ref.
Pivot Table Cache Definition	Pivot Table, Workbook	pivotCacheDefinition	§12.3.12
Pivot Table Cache Records	Pivot Table Cache Definition	pivotCacheRecords	§12.3.13
Query Table	Worksheet	queryTable	§12.3.14
Shared String Table	Workbook	sst	§12.3.15
Shared Workbook Revision Headers	Workbook	headers	§12.3.16
Shared Workbook Revision Log	Shared Workbook Revision Headers	revisions	§12.3.17
Shared Workbook User Data	Workbook	users	§12.3.18
Single Cell Table Definitions	Dialogsheet, Worksheet	singleXmlCells	§12.3.19
Styles	Workbook	styleSheet	§12.3.20
Table Definition	Dialogsheet, Worksheet	table	§12.3.21
Volatile Dependencies	Workbook	volTypes	§12.3.22
Workbook	SpreadsheetML package	workbook	§12.3.23
Worksheet	Workbook	worksheet	§12.3.24

end note]

12.3.1 Calculation Chain Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.calcChain+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/calcChain

An instance of this part type contains an ordered set of references to all cells in all worksheets in the workbook whose value is calculated from any formula. The ordering allows inter-related cell formulas to be calculated in the correct order when a worksheet is loaded for use.

A package shall contain no more than one Calculation Chain part. If it exists, that part shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains a relationship to the Calculation Chain part, which is stored in the ZIP item calcChain.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId7"
    Type="http://…/calcChain" Target="calcChain.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be calcChain.

[*Example*: Cells D8, E8, and F8 each contain a value that is the result of calculations that shall be performed in the order E8, D8, F8:

```
<calcChain xmlns="…">
  <c r="E8" i="1"/>
  <c r="D8"/>
  <c r="F8" s="1"/>
</calcChain>
```

end example]

A Calculation Chain part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Calculation Chain part shall not have implicit or explicit relationships to any part defined by ISO/IEC 29500.

12.3.2 Chartsheet Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.chartsheet+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/chartsheet

An instance of this part type represents a chart that is stored in its own sheet.

A package is permitted to contain zero or more Chartsheet parts. Each such part shall be the target of an explicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains three relationships to Chartsheet parts, which are stored in the ZIP items chartsheets/sheet*N*.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/chartsheet" Target="chartsheets/sheet1.xml"/>
  <Relationship Id="rId5"
    Type="http://…/chartsheet" Target="chartsheets/sheet2.xml"/>
  <Relationship Id="rId6"
    Type="http://…/chartsheet" Target="chartsheets/sheet3.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be chartsheet.

[*Example*: sheet1.xml refers to a drawing that is the target of a relationship in the Chartsheet part's relationship item:

```
<chartsheet xmlns:r="…" …>
  <sheetViews>
    <sheetView scale="64"/>
  </sheetViews>
  <drawing r:id="rId1"/>
</chartsheet>
```

end example]

A Chartsheet part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Chartsheet part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Printer Settings (§15.2.15)

A Chartsheet part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Drawings (§12.3.8)

A Chartsheet part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.3 Comments Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.comments+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/comments

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

An instance of this part type contains all the comments for a given worksheet, as well as the names of the authors of those comments.

A package shall contain exactly one Comments part for each worksheet that contains one or more comments. If a Comments part exists, it shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Worksheet part-relationship item contains a relationship to the Comments part, which is stored in the ZIP item comments2.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/comments" Target="../comments2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be comments.

[*Example*: This Comments part results from a workbook that has one or more comments from each of two people: James Jones and Mary Smith:

```
<comments xmlns:st="…" >
  <authors>
    <author>James Jones</author>
    <author>Mary Smith</author>
  </authors>
  <commentList>
    <comment r="C7" authorId="0">
      <text>
        <st:r>
          <st:rPr>
            …
          </st:rPr>
          <st:t>James Jones:</st:t>
        </st:r>
        <st:r>
          <st:rPr>
            …
          </st:rPr>
          <st:t>Check that this date is correct.</st:t>
        </st:r>
      </text>
    </comment>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <comment r="E7" authorId="1">
        …
      </comment>
    </commentList>
  </comments>
```

end example]

A Comments part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Comments part shall not implicit or explicit relationships to any part defined by ISO/IEC 29500.

12.3.4 Connections Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.connections+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/connections

An instance of this part type describes all of the connections currently established for a workbook.

A package shall contain no more than one Connections part, and that part shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains a relationship to the Connections part, which is stored in the ZIP item connections.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/connections" Target="connections.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be connections.

[*Example*: A workbook has three connections, two from one worksheet, and one from another. connections.xml defines these three connections:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<connections …>
  <connection id="1" odcFile="…" keepAlive="1" name="…" type="5"
    refreshedVersion="2" background="1" saveData="1">
    <dbPr connection="Provider=MSDASQL.1;Persist Security Info=True;Data
Source=dBASE Files;Extended Properties=&amp;quot;DSN=dBASE Files;DBQ=E:\MY
DOCUMENTS;DefaultDir=E:\MY
DOCUMENTS;DriverId=533;MaxBufferSize=2048;PageTimeout=5;&amp;quot;;Initial
Catalog=E:\MY DOCUMENTS" command="`E:\MY DOCUMENTS`\`ADDRESS`" commandType="3"/>
  </connection>
  <connection id="2" …>
    <dbPr … />
  </connection>
  <connection id="3" …>
    <dbPr … />
  </connection>
</connections>
```

end example]

A Connections part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Connections part shall not have implicit or explicit relationships to any part defined by ISO/IEC 29500..

12.3.5 Custom Property Part

Content Type:	Any content, support for which is application-defined. [*Note*: Some examples of formats which might be supported include: • application/vnd.openxmlformats-officedocument.spreadsheetml.customProperty • application/xml *end note*]
Root Namespace:	Not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customProperty

This part supports the storage of user-defined data.

[*Note*: It is recommended that a Custom Property Part contain XML content for improved interoperability; however, there is no requirement on the format of the content contained in a Custom Property Part. *end note*]

A package is permitted to contain zero or more Custom Property parts, and each such part shall be the target of an implicit relationship from the Worksheet (§12.3.24) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following Worksheet part-relationship item contains a relationship to the Custom Property part, which is stored in the ZIP item CustomProperty.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId7"
    Type="http://…/customProperty" Target="CustomProperty.xml"/>
</Relationships>
```

where the contents of CustomProperty.xml contain the following XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CustomApplicationData xmlns="…">
    <CustomProperty name="PropertyName" value="PropertyValue" />
</CustomApplicationData>
```

end example]

A Custom Property part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Custom Property part shall not have implicit or explicit relationships to any part defined by ISO/IEC 29500.

12.3.6 Custom XML Mappings Part

Content Type:	application/xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/xmlMaps

An instance of this part type contains a schema for an XML file, and information on the behavior that is used when allowing this custom XML schema to be mapped into the spreadsheet.

A package shall contain no more than one Custom XML Mappings part, and that part shall be the target of an implicit relationship from the Workbook part (§12.3.23). The Worksheet part into which this data is imported shall also have a relationship file that targets one or more Table Definition (§12.3.21) parts and/or one or more Single Cell Table Defintions (§12.3.19) parts.

[*Example*: The following Workbook part-relationship item contains a relationship to the Custom XML Mappings part, which is stored in the ZIP item xmlMaps.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId9"
    Type="http://…/xmlMaps" Target="xmlMaps.xml"/>
</Relationships>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

The root element for a part of this content type shall be MapInfo.

[*Example*: xmlMaps.xml contains the following:

```
<mapInfo SelectionNamespaces="">
  <Schema ID="Schema1">
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element nillable="true" name="names">
            <xsd:complexType>
           <xsd:sequence minOccurs="0">
             <xsd:element minOccurs="0" maxOccurs="unbounded"
               nillable="true" name="name" form="unqualified">
               <xsd:complexType>
                 <xsd:sequence minOccurs="0">
                   <xsd:element minOccurs="0" nillable="true"
                     type="xsd:string" name="firstname"
                     form="unqualified"/>
                   <xsd:element minOccurs="0" nillable="true"
                     type="xsd:string" name="initial"
                     form="unqualified"/>
                   <xsd:element minOccurs="0" nillable="true"
                     type="xsd:string" name="lastName"
                     form="unqualified"/>
                 </xsd:sequence>
               </xsd:complexType>
             </xsd:element>
           </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </Schema>
  <Map ID="1" Name="names_Map" RootElement="names" SchemaID="Schema1"
    ShowImportExportValidationErrors="false" AutoFit="true"
    Append="false"
    PreserveSortAFLayout="true" PreserveFormat="true">
    <DataBinding FileBinding="Test.xml" DataBindingLoadMode="1"/>
  </Map>
</mapInfo>
```

end example]

A Custom XML Mappings part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A Custom XML Mappings part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.7 Dialogsheet Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.dialogsheet+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/dialogsheet

An instance of this part type contains information about a legacy custom dialog box for a user form.

A package is permitted to contain one or more Dialogsheet parts, and each such part shall be the target of an explicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains relationships to a Dialogsheet part, which is stored in the ZIP item dialogsheets/sheet1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/dialogsheet" Target="dialogsheets/sheet1.xml"/>
</Relationships>
```

The Workbook part contains the following:

```
<workbook xmlns:r="…" …>
  …
  <sheets>
    …
    <sheet name="Dialog1" sheetId="4" r:id="rId2"/>
  </sheets>
  …
</workbook>
```

end example]

The root element for a part of this content type shall be dialogsheet.

[*Example*: sheet1.xml contains the following:

```
<dialogsheet xmlns:r="…" …>
  <sheetPr>
    <pageSetUpPr/>
  </sheetPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <sheetViews>
      …
    </sheetViews>
      …
    <legacyDrawing r:id="rId1"/>
  </dialogsheet>
```

end example]

A Dialogsheet part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Dialogsheet part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Printer Settings (§15.2.15)

A Dialogsheet part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Embedded Control Persistence (§15.2.9)
- Drawings (§12.3.8)
- Embedded Object (§15.2.10)

A Dialogsheet part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.8 Drawings Part

Content Type:	application/vnd.openxmlformats-officedocument.drawing+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/drawing

An instance of this part type contains the presentation and layout information for one or more drawing elements that are present on this worksheet.

A package is permitted to contain one or more Drawings parts, and each such part shall be the target of an explicit relationship from a Worksheet part (§12.3.24), or a Chartsheet part (§12.3.2). There shall be only one Drawings part per worksheet or chartsheet.

[*Example*: The following Chartsheet part-relationship item contains a relationship to a Drawings part, which is stored in the ZIP item ../drawings/drawing1.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http:// …/drawing" Target="../drawings/drawing1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be wsDr.

[*Example*: drawing1.xml refers to a chart that is the target of a relationship in the Drawing part's relationship item:

```
<xdr:wsDr xmlns:xdr="…" xmlns:a="…">
  <xdr:absoluteAnchor>
    <xdr:pos x="1518046" y="-1443632"/>
    <xdr:extents cx="8587382" cy="5848945"/>
    <xdr:graphicFrame macro="">
      <xdr:nvGraphicFramePr>
        <xdr:cNvPr id="24" name="Chart 24" descr=""/>
        <xdr:cNvGraphicFramePr/>
      </xdr:nvGraphicFramePr>
      <xdr:xfrm>
        <a:off x="0" y="0"/>
        <a:ext cx="0" cy="0"/>
      </xdr:xfrm>
      <a:graphic>
        <a:graphicData uri="http://…/chart">
          <a:chart relId="rId1"/>
        </a:graphicData>
      </a:graphic>
    </xdr:graphicFrame>
    <xdr:clientData/>
  </xdr:absoluteAnchor>
</xdr:wsDr>
```

end example]

A Drawings part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Drawings part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Chart (§14.2.1)
- Content Part (§15.2.4)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Hyperlinks (§15.3)
- Images (§15.2.14)

A Drawings part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.9 External Workbook References Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.externalLink+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/externalLink

An instance of this part specifies information about data referenced in other SpreadsheetML packages.

[*Example*: Consider two workbooks, wb1 and wb2, stored in packages called wb1.xlsx and wb2.xlsx, respectively. The value of a cell on a worksheet in wb1 can be computed using the value of one or more cells in wb2. This is done by having wb1 contain an external reference to wb2. *end example*]

A package is permitted to contain one or more External Workbook References parts, and those parts shall be the target of an explicit relationship in the Workbook part (§12.3.23).

[*Example*: A Workbook part for wb1 contains the following, which indicates that somewhere in its three worksheets, an external reference is made to a target specified in relationship id rId4 of the part's relationship item:

```
<workbook xmlns:r="…"/>
  …
  <sheets>
    <sheet name="Sheet1" sheetId="1" r:id="rId1"/>
    <sheet name="Sheet2" sheetId="2" r:id="rId2"/>
    <sheet name="Sheet3" sheetId="3" r:id="rId3"/>
  </sheets>
  …
  <externalReferences>
    <externalReference r:id="rId4"/>
  </externalReferences>
  …
</workbook>
```

That part's relationship item contains the following:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/externalLink"
    Target="externalReferences/externalReference1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be externalLink.

[*Example*: externalReference1.xml contains:

```
<externalLink xmlns:r="…" … r:id="rId1">
  <externalBook>
    <sheetNames>
      <sheetName val="Sheet1"/>
      <sheetName val="Sheet2"/>
      <sheetName val="Sheet3"/>
    </sheetNames>
    <sheetDataSet>
      <sheetData sheetId="0">
        <row r="7">
          <cell r="C8">
            <v>0</v>
          </cell>
        </row>
      </sheetData>
      <sheetData sheetId="1"/>
      <sheetData sheetId="2"/>
    </sheetDataSet>
  </externalBook>
</externalLink>
```

This part's relationship item contains the following:

```
<Relationships …>
  <Relationship Id="rId1"
    Type="…/externalReference"
    Target="wb2.xlsx" TargetMode="External"/>
</Relationships>
```

where wb2.xlsx is the workbook in which one or more cells' values are used in calculating the values of a cell in workbook wb1. *end example*]

An External Workbook References part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

An External Workbook References part shall specify an explicit relationship to one or more External Workbooks (§12.4).

An External Workbook References part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

12.3.10 Metadata Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.sheetMetadata+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/sheetMetadata

An instance of this part type contains information relating to a cell whose value is related to one or more other cells via OnLine Analytical Processing (OLAP) technology.

A package shall contain no more than one Cell Metadata part, and that part shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains a relationship to the Metadata part, which is stored in the ZIP item metadata.xml. Cell B3 contains the formula `CUBEMEMBER ("externalData", "[Account].[All Account]")`:

```
<Relationships xmlns="…">
  <Relationship Id="rId10"
    Type="http://…/sheetMetadata" Target="metadata.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be metadata.

[*Example*: metadata.xml contains the following:

```
<metadata …>
  <metadataTypes count="1">
    <metadataType name="XLMDX" minSupportedVersion="120000" copy="1"
      pasteAll="1" pasteValues="1" merge="1" splitFirst="1" rowColShift="1"
      clearFormats="1" clearComments="1" assign="1" coerce="1"/>
  </metadataTypes>
  <metadataStrings count="2">
    <s v="externalData"/>
    <s v="[Account].[All Account]"/>
  </metadataStrings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <mdxMetadata count="1">
      <m n="0" f="m">
        <t c="1">
          <n v="1"/>
        </t>
      </m>
    </mdxMetadata>
    <valueMetadata count="1">
      <b>
        <r t="1" v="0"/>
      </b>
    </valueMetadata>
  </metadata>
```

The corresponding Connections part contains the following:

```
  <connections …>
    <connection id="1" odcFile="…" keepAlive="1" name="externalData"
      description="…" type="5" refreshedVersion="3" background="1">
      <dbPr connection="Provider=MSOLAP.2;…" command="Budget" commandType="1"/>
      <olapPr sendLocale="1" rowDrillCount="1000" serverFill="1"
        serverNumberFormat="1" serverFont="1" serverFontColor="1"/>
    </connection>
  </connections>
```

The corresponding Volatile Dependencies part contains the following:

```
  <volTypes …">
    <volType type="cubeFunctions">
      <main first="externalData">
        <tp t="e">
          <v>#N/A</v>
          <stp>1</stp>
          <r r="B3" s="1"/>
        </tp>
      </main>
    </volType>
  </volTypes>
```

The corresponding Pivot Table Cache Definition part contains the following:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<pivotCacheDefinition … saveData="0" refreshedBy="…"
  refreshedDateIso="2005-11-28T16:55:44" backgroundQuery="1" createdVersion="3"
  refreshedVersion="3" recordCount="0">
  <cacheSource type="external" connectionID="1"/>
  <cacheFields count="0"/>
  <cacheHierarchies count="6">
    …
  </cacheHierarchies>
  <kpis count="0"/>
  <tupleCache>
    <queryCache count="3">
      <query mdx="[product].[category]"/>
      <query mdx=""/>
      <query mdx="[Account].[All Account]">
        <tpls c="1">
          <tpl hier="0" item="4294967295"/>
        </tpls>
      </query>
    </queryCache>
  </tupleCache>
</pivotCacheDefinition>
```

end example]

A Metadata part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Metadata part shall not have implicit or explicit relationships to any part defined by ISO/IEC 29500.

12.3.11 Pivot Table Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.pivotTable+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/pivotTable

An instance of this part type contains a pivot table definition.

A package shall contain exactly one Pivot Table part per pivot table, and each such part shall be the target of an implicit relationship in the relationship part for the Worksheet part (§12.3.24) that corresponds to the worksheet containing the pivot table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: The following Worksheet part-relationship item contains a relationship to two Pivot Table parts, which are stored in the ZIP items ../pivotTables/pivotTable*N*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/pivotTable" Target="../pivotTables/pivotTable1.xml"/>
  <Relationship Id="rId2"
    Type="http://…/pivotTable" Target="../pivotTables/pivotTable2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be pivotTableDefinition.

[*Example*: pivotTable1.xml contains the following:

```
<pivotTableDefinition … cache="4" applyNumberFormats="0" applyBorderFormats="0"
  applyFontFormats="0" applyPatternFormats="0" applyAlignmentFormats="0"
  applyWidthHeightFormats="1" dataCaption="Data" updatedVersion="3"
  minRefreshableVersion="3" useAutoFormatting="1" itemPrintTitles="1"
  createdVersion="3" indent="0" outline="1" outlineData="1">
  <location ref="H4:H5" firstHeaderRow="1" firstDataRow="1" firstDataCol="0"/>
  <pivotFields count="1">
    <pivotField dataField="1" numFmtId="0" outline="1"
      subtotalTop="1" showAll="0" measureFilter="0" sortType="manual"/>
  </pivotFields>
  <rowItems count="1">
    <i t="data"/>
  </rowItems>
  <colItems count="1">
    <i t="data"/>
  </colItems>
  <dataFields count="1">
    <dataField name="Sum of 1000" fld="0" subtotal="average"
      baseField="0" baseItem="0" numFmtId="0"/>
  </dataFields>
  <tableStyle name="TableStyle2" showRowHeaders="1" showColHeaders="1"
    showRowStripes="1" showColStripes="1"/>
</pivotTableDefinition>
```

end example]

A Pivot Table part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Pivot Table part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Pivot Table Cache Definition (§12.3.12).

A Pivot Table part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

12.3.12 Pivot Table Cache Definition Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.pivotCacheDefinition+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/pivotCacheDefinition

An instance of this part type contains a cache definition for a pivot table.

A package shall contain exactly one Pivot Table Cache Definition part per pivot table, and each such part shall be the target of an implicit relationship from a Pivot Table (§12.3.11) part as well as an explicit relationship from a Workbook (§12.3.23) part.

[*Example*: The following Pivot Table part-relationship item contains a relationship to the Pivot Table Cache Definition part, which is stored in the ZIP item ../pivotCache/pivotCacheDefinition2.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type=http://…/pivotCacheDefinition
    Target="../pivotCache/pivotCacheDefinition2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be pivotCacheDefinition.

[*Example*: pivotCacheDefinition2.xml contains the following:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<pivotCacheDefinition … r:id="rId1" refreshedBy="John Jones"
  refreshedDateIso="2005-11-18T16:47:49" createdVersion="3"
  refreshedVersion="3" recordCount="11">
  <cacheSource type="worksheet">
    <worksheet range="C4:C15" sheet="Sheet1"/>
  </cacheSource>
  <cacheFields count="1">
    <cacheField name="1000">
      <sharedItems containsSemiMixedTypes="0" containsString="0"
        containsNumber="1" containsInteger="1" minValue="234
        maxValue="2543"/>
    </cacheField>
  </cacheFields>
</pivotCacheDefinition>
```

end example]

A Pivot Table Cache Definition part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Pivot Table Cache Definition part is permitted to have an explicit relationship to the following part defined by ISO/IEC 29500:

- Pivot Table Cache Records (§12.3.13).

A Pivot Table Cache Definition part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

12.3.13 Pivot Table Cache Records Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.pivotCacheRecords+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/pivotCacheRecords

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An instance of this part type contains the cache records for a pivot table.

A package shall contain zero or one Pivot Table Cache Records part per pivot table, and each such part shall be the target of an explicit relationship in the Pivot Table Cache Definition (§12.3.12) part for the corresponding pivot table.

[*Example*: The following Pivot Table Cache Definition part-relationship item contains a relationship to the Pivot Table Cache Records part, which is stored in the ZIP item pivotCacheRecords2.xml:

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/pivotCacheRecords" Target="pivotCacheRecords2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be pivotCacheRecords.

[*Example*: pivotCacheRecords2.xml contains the following:

```
<pivotCacheRecords … count="11">
  <r>
    <n v="1234"/>
  </r>
    …
  <r>
    <n v="876"/>
  </r>
</pivotCacheRecords>
```

end example]

A Pivot Table Cache Records part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Pivot Table Cache Records part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.14 Query Table Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.queryTable+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/queryTable

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An instance of this part type contains information that describes how the source table is connected to an external data source, and defines the properties that is used when this table is refreshed from that source.

A package is permitted to contain one Query Table part per table, and each of those parts shall be the target of an implicit relationship from the corresponding Table Definitions (§12.3.21) part.

©ISO/IEC 2016 – All rights reserved

[*Example*: The following Table part-relationship item contains a relationship to the Query Table part corresponding to the connections details for that table. These parts are stored in the ZIP items ../queryTables/queryTable*n*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/queryTable"
    Target="../queryTables/queryTable1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be queryTable.

[*Example*: queryTable2.xml deals with a connection to a database file having the seven fields shown:

```
<queryTable … name="+Connect to New Data Source_1"
  growShrinkType="insertDelete" connectionId="2" autoFormatId="16"
  applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="1"
  applyPatternFormats="1" applyAlignmentFormats="0"
  applyWidthHeightFormats="0">
  <queryTableRefresh nextId="8">
    <queryTableFields count="7">
      <queryTableField id="1" name="ACCOUNT"/>
      <queryTableField id="2" name="CHECKNUM"/>
      <queryTableField id="3" name="DATE"/>
      <queryTableField id="4" name="AMOUNT"/>
      <queryTableField id="5" name="PAYEE"/>
      <queryTableField id="6" name="CHARGECODE"/>
      <queryTableField id="7" name="DESCRIPT"/>
    </queryTableFields>
  </queryTableRefresh>
</queryTable>
```

end example]

A Query Table part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Query Table part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.15 Shared String Table Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.sharedStrings+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/sharedStrings

An instance of this part type contains one occurrence of each unique string that occurs on all worksheets in a workbook.

A package shall contain exactly one Shared String Table part, and that part shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared String Table part, which is stored in the ZIP item sharedStrings.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId6"
    Type="http://…/sharedStrings" Target="sharedStrings.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be sst.

[*Example*: Here are three of the six strings used in the worksheets:

```
<sst xmlns:st="…" … totalCount="6" uniqueCount="6">
  <sstItem>
    <t>Expenses Log</t>
  </sstItem>
  <sstItem>
    <t>Period Start</t>
  </sstItem>
  <sstItem>
    <t>Period End</t>
  </sstItem>
  …
</sst>
```

end example]

A Shared String Table part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Shared String Table part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

12.3.16 Shared Workbook Revision Headers Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.revisionHeaders+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/revisionHeaders

An instance of this part type contains information about each of the editing sessions performed on the parent workbook at the worksheet level (worksheets added and rearranged in each session).

A package shall contain at most one Shared Workbook Revision Headers part. If it exists, that part shall be the target of an implicit relationship from the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared Workbook Revision Headers part, which is stored in the ZIP item handout revisions/revisionHeaders.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId9"
    Type="http://…/revisionHeaders"
    Target="revisions/revisionHeaders.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be headers.

[*Example*: revisionHeaders.xml contains the following:

```
<headers xmlns:r="…" guid="{233BEE23-EB5C-4542-905D-0230EFFED88B}"
  diskRevisions="1" revisionId="4" version="3">
  <header guid="…" dateTime="…" maxSheetId="4" userName="…" r:id="rId1">
    <sheetIdMap count="3">
      …
    </sheetIdMap>
  </header>
  …
  <header guid="…" dateTime="…" maxSheetId="4" userName="…" r:id="rId3">
    <sheetIdMap count="3">
      …
    </sheetIdMap>
  </header>
</headers>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A Shared Workbook Revision Headers part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Shared Workbook Revision Headers part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Shared Workbook Revision Log (§12.3.17)

A Shared Workbook Revision Headers part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

12.3.17 Shared Workbook Revision Log Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.revisionLog+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/revisionLog

An instance of this part type contains information about edits performed on individual cells in the parent workbook's worksheets in each editing session.

A package shall contain one Shared Workbook Revision Log part for each session's set of changes, and those parts shall be the target of an explicit relationship from the Shared Workbook Revision Headers (§12.3.16) part.

[*Example*: The following Shared Workbook Revision Headers part-relationship item contains a number of relationships to Shared Workbook Revision Log parts, which are stored in the ZIP item revisionLog*N*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/revisionLog" Target="revisionLog1.xml"/>
  …
  <Relationship Id="rId6"
    Type="http://…/revisionLog" Target="revisionLog6.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be revisions.

[*Example*: revisionLog2.xml contains the following:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<revisions xmlns:xs="…" …>
  <rfmt sheetId="1" sqref="B4:B15">
    <dxf>
      <xs:fill>
        <xs:pattern patternType="solid">
          <xs:fgColor type="icv" val="64"/>
          <xs:bgColor type="rgb" val="4278252287"/>
        </xs:pattern>
      </xs:fill>
    </dxf>
  </rfmt>
  <rcv guid="{CBCE5672-5A4D-48C9-A120-F72804F8CF64}" action="delete"/>
  <rcv guid="{CBCE5672-5A4D-48C9-A120-F72804F8CF64}" action="add"/>
</revisions>
```

end example]

A Shared Workbook Revision Log part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Shared Workbook Revision Log part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.18 Shared Workbook User Data Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.userNames+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/usernames

An instance of this part type contains a list of all the users that are sharing the parent workbook.

A package shall contain at most one Shared Workbook User Data part, and that part shall be the target of an implicit relationship in the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared Workbook User Data part, which is stored in the ZIP item revisions/userNames.xml:

```
<Relationships xmlns="…">
  Relationship Id="rId8"
    Type="http://…/usernames" Target="revisions/userNames.xml"/>
</Relationships>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The root element for a part of this content type shall be users.

[*Example*: userNames.xml shows that there are two users sharing this workbook:

```
<users … count="2">
  <usrinfo guid="{B5A024F7-40BE-4A48-9B6D-B1655241C84D}"
    name="Mary Jones" id="-264292310" dateTime="2005-11-18T18:53:16"/>
  <usrinfo …/>
</users>
```

end example]

A Shared Workbook User Data part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Shared Workbook User Data part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.19 Single Cell Table Definitions Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.tableSingleCells+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/tableSingleCells

An instance of this part type contains information on how to map non-repeating elements from a custom XML file into cells in a worksheet. [*Note*: Repeating custom XML elements are mapped using a Table (§12.3.21). *end note*]

A package shall contain at most one Single Cell Table Definitions part per worksheet, and that part shall be the target of an implicit relationship from a Worksheet (§12.3.24) part. A Single Cell Table Definitions part can describe one or more single cell table definitions for any given worksheet.

[*Example*: The following Worksheet part-relationship item contains a relationship to the Single Cell Table Definitions part, which is stored in the ZIP item ../tables/tableSingleCells1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/tableSingleCells"
    Target="../tables/tableSingleCells1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be singleXmlCells.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: A worksheet contains two single cell table definitions; e.g., ../tables/tableSingleCells1.xml contains the following, where the elements id and count are nested inside element names:

```
<singleXmlCells …>
  <singleCell id="1" name="Table1" displayName="Table1" ref="B4">
    <cellPr id="1" uniqueName="id">
      <xmlPr mapId="1" xpath="/names/id" xmlDataType="string"/>
    </cellPr>
  </singleCell>
  <singleCell id="2" name="Table2" displayName="Table2" ref="B7">
    <cellPr id="1" uniqueName="count">
      <xmlPr mapId="1" xpath="/names/count" xmlDataType="integer"/>
    </cellPr>
  </singleCell>
</singleXmlCells>
```

end example]

A Single Cell Table Definitions part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Single Cell Table Definitions part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.20 Styles Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.styles+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/styles

An instance of this part type contains all the characteristics for all the cells in the workbook. Such information includes numeric and text formatting, alignment, font, color, and border.

A package shall contain no more than one Styles part, and that part shall be the target of an implicit relationship from the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Styles part, which is stored in the ZIP item styles.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/styles" Target="styles.xml"/>
</Relationships>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

The root element for a part of this content type shall be styleSheet.

[*Example*:

```
<styleSheet xmlns="…">
  <numFmts count="5">
    <numFmt numFmtId="164" formatCode="&quot;$&quot;#,##0.00"/>
    <numFmt numFmtId="165"
      formatCode="&quot;Yes&quot;;&quot;Yes&quot;;&quot;No&quot;"/>
    <numFmt numFmtId="166"
      formatCode="&quot;True&quot;;&quot;True&quot;;&quot;False&quot;"/>
    <numFmt numFmtId="167"
      formatCode="&quot;On&quot;;&quot;On&quot;;&quot;Off&quot;"/>
    <numFmt numFmtId="168"
      formatCode="[$€-2]\ #,##0.00_);[Red]\([$€-2]\ #,##0.00\)"/>
  </numFmts>
  <fonts count="5">
    …
  </fonts>
  <fills count="4">
    …
  </fills>
  <borders count="1">
    …
  </borders>
  …
  <colors>
    …
  </colors>
</styleSheet>
```

end example]

A Styles part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Styles part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.21 Table Definition Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.table+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/table

An instance of this part type contains a description of a single table and its autofilter information. (The data for the table is stored in the corresponding Worksheet part.)

A package shall contain one Table Definition part per table, and each such part shall be the target of an implicit relationship from the Worksheet (§12.3.24) part that corresponds to the worksheet containing that table.

[*Example*: The following Worksheet part-relationship item contains relationships to two Table Definition parts, which are stored in the ZIP items ../tables/table*N*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/table" Target="../tables/table1.xml"/>
  <Relationship Id="rId3"
    Type="http://…/table" Target="../tables/table2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be table.

[*Example*: table2.xml describes a table that spans a 2-column range of cells, F2:G19:

```
<table xmlns:af="…" … id="2" name="Table2" displayName="Table2" ref="F2:G19"
  totalsRowShown="0" headerRowDxfId="7">
  <autoFilter ref="F2:G19"/>
  <tableColumns count="2">
    <tableColumn id="1" name="Salesman" dataDxfId="9" totalsRowDxfId="6"/>
    <tableColumn id="2" name="Units" dataDxfId="8" totalsRowDxfId="5"/>
  </tableColumns>
  <tableStyle name="TableStyle2" showFirstColumn="0" showLastColumn="0"
    showRowStripes="1" showColumnStripes="1"/>
</table>
```

When the filter "Salesman equal to Smith" is applied, the autoFilter element in table2.xml is extended, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<autoFilter ref="F2:G19">
  <af:filterColumn colId="0">
    <af:filters>
      <af:filter val="Smith"/>
    </af:filters>
  </af:filterColumn>
</autoFilter>
```

end example]

A Table Definition part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Table Definition part is permitted to explicit relationships to the following parts defined by ISO/IEC 29500:

- Query Table (§12.3.14)

A Table Definition part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.22 Volatile Dependencies Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.volatileDependencies+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/volatileDependencies

An instance of this part type contains information involving real-time data formulas in a workbook. Real-time data formulas return values that change over time — often every few seconds — and require connectivity to programs outside of the workbook to retrieve their data. In cases where those programs are not available, real-time data formulas can use the information stored in the Volatile Dependencies part to calculate results, rather than generate errors. More information on real-time data functions can be found in §18.17.7.284 and §18.17.7.65 through §18.17.7.71.

A package shall contain exactly one Volatile Dependencies part, and that part shall be the target of an implicit relationship from the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Volatile Dependencies part, which is stored in the ZIP item volatileDependencies.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<Relationships xmlns="…">
  <Relationship Id="rId8"
    Type="http://…/volatileDependencies"
    Target="volatileDependencies.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be volTypes.

[*Example*:

```
<volTypes xmlns="…"/>
```

end example]

A Volatile Dependencies part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Volatile Dependencies part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.23 Workbook Part

Content Type(s):	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet.main+xml application/vnd.openxmlformats-officedocument.spreadsheetml.template.main+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument

An instance of this part type contains workbook data and references to all of its worksheets.

A package shall contain exactly one Workbook part, and that part shall be the target of a relationship in the package-relationship item.

[*Example*: The following SpreadsheetML package-relationship item contains a relationship to the Workbook part, which is stored in the ZIP item xl/workbook.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="xl/workbook.xml"/>
</Relationships>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The root element for a part of this content type shall be workbook.

[*Example*: This workbook has three worksheets, named January, February, and March:

```
<workbook xmlns="…" xmlns:r="…">
  <fileVersion lastEdited="4" lowestEdited="4" rupBuild="3417"/>
  <bookViews>
    <workbookView xWindow="240" yWindow="15" windowWidth="8505"
      windowHeight="6240"/>
  </bookViews>
  <sheets>
    <sheet name="January" sheetId="1" r:id="rId1"/>
    <sheet name="February" sheetId="2" r:id="rId2"/>
    <sheet name="March" sheetId="3" r:id="rId3"/>
  </sheets>
  <workbookPr showObjects="all"/>
  <webPublishing codePage="1252"/>
  <calcPr calcId="122211" fullCalcOnLoad="1"/>
</workbook>
```

end example]

A Workbook part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Workbook part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Calculation Chain (§12.3.1)
- Cell Metadata (§12.3.10)
- Connections (§12.3.4)
- Custom XML Mappings (§12.3.6)
- Custom XML Data Storage (§15.2.4)
- Shared String Table (§12.3.15)
- Shared Workbook Revision Headers (§12.3.16)
- Shared Workbook User Data (§12.3.18)
- Styles (§12.3.20)
- Theme (§14.2.7)
- Thumbnail (§15.2.16)
- Volatile Dependencies (§12.3.22)

A Workbook part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- Chartsheet (§12.3.2)
- Dialogsheet (§12.3.7)
- External Workbook References (§12.3.8)
- Pivot Table Cache Definition (§12.3.12)
- Worksheet (§12.3.24)

A Workbook part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.3.24 Worksheet Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml
Root Namespace:	http://purl.oclc.org/ooxml/spreadsheetml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/worksheet

An instance of this part type contains all the data, formulas, and characteristics associated with a given worksheet.

A package shall contain exactly one Worksheet part per worksheet, and those parts shall be the target of an explicit relationship from the Workbook (§12.3.23) part. Specifically, the id attribute on the sheet element shall reference the desired worksheet part.

[*Example*: The following Workbook part-relationship item contains three relationships to Worksheet parts, which are stored in the ZIP items worksheets/sheet*N*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/worksheet" Target="worksheets/sheet1.xml"/>
  <Relationship Id="rId2"
    Type="http://…/worksheet" Target="worksheets/sheet2.xml"/>
  <Relationship Id="rId3"
    Type="http://…/worksheet" Target="worksheets/sheet3.xml"/>
 </Relationships>
```

end example]

The root element for a part of this content type shall be worksheet.

[*Example*: This worksheet, has cells in the range B1 to F8. Row 8 contains three cells whose values are calculated using the following formulas: D8=SUM(D5:D7), E8=SUM(E5:E7), and F8= D8-E8:

```
<worksheet xmlns="…" …>
  <sheetPr/>
  <dimension range="B1:F8"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
…
<sheetData>
  <row r="1" spans="2:6" ht="360">
    <c r="B1" s="1" t="s">
      <v>0</v>
    </c>
  </row>
  …
  <row r="8" spans="2:6" ht="360">
    <c r="D8" s="5">
      <f>SUM(D5:D7)</f>
      <v>2280.5299999999997</v>
    </c>
    <c r="E8" s="5">
      <f>SUM(E5:E7)</f>
      <v>1251.31</v>
    </c>
    <c r="F8" s="6">
      <f>D8-E8</f>
      <v>1029.2199999999998</v>
    </c>
  </row>
</sheetData>
…
</worksheet>
```

end example]

A Worksheet part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Worksheet part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Comments (§12.3.3)
- Pivot Table Definitions (§12.3.11)
- Printer Settings (§15.2.15)
- Query Table Part (§12.3.14)
- Single Cell Table Definitions (§12.3.19)
- Table Definition (§12.3.21)

A Worksheet part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

- Drawings (§12.3.8)
- Embedded Control Persistence (§15.2.9)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlinks (§15.3)
- Images (§15.2.14)

A Worksheet part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

12.4 External Workbooks

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/externalLinkPath

An *external workbook* is a SpreadsheetML package whose contents are being referenced by the current SpreadsheetML package. When a package refers to external workbooks, it shall store the location of those workbooks using this relationship.

A package is permitted to contain one or more External Workbook relationships, and those relationships shall be an explicit relationship from the External Workbook References (§12.3.9) part.

[*Example*: An External Workbook References part, which references the package c:\sourceData.xlsx would have an External Workbook relationship, which points at that file:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/externalLinkPath"
    Target="c:\sourceData.xlsx" TargetMode="External"/>
</Relationships>
```

end example]

A external workbook shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

13. PresentationML

This clause contains specifications for relationship items and parts that are specific to PresentationML. Parts than can occur in a PresentationML document, but are not PresentationML-specific, are specified in §15.2. Unless stated explicitly, all references to relationship items, content-type items, and parts in this clause refer to PresentationML ZIP items.

13.1 Glossary of PresentationML-Specific Terms

The following terms are used in the context of a PresentationML document:

handout — A printed set of slides that can be handed out to an audience for future reference.

note — A slide annotation, reminder, or piece of text intended for the presenter or the audience.

presentation — A collection of slides intended to be viewed by an audience.

slide — A frame containing one or more pieces of text and/or images.

slide layout — The organization of elements on a slide.

13.2 Package Structure

A PresentationML package shall contain a package-relationship item and a content-type item. The package-relationship item shall have implicit relationships with targets of the following type:

- One Presentation part (§13.3.6).

The package-relationship item is permitted to have implicit relationships with targets of the following type:

- Digital Signature Origin (§15.2.7)
- File Property parts (§15.2.12) (Application-Defined File Properties, Core File Properties, and Custom File Properties), as appropriate.
- Thumbnail (§15.2.16).

The required and optional relationships between parts are defined in §13.3 and its subordinate clauses.

[*Example*: The following package represents the minimal conformant PresentationML package as defined by ISO/IEC 29500:

First, the content type for relationship parts and the Main Presentation part (the only required part) must be defined (physically located at /[Content_Types].xml in the package):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<Types xmlns="…">
  <Default Extension="rels"
    ContentType="application/vnd.openxmlformats-
      package.relationships+xml"/>
  <Override PartName="/presentation.xml"
    ContentType="application/vnd.openxmlformats-
      officedocument.presentationml.presentation.main+xml"/>
</Types>
```

Next, the single required relationship (the package-level relationship to the Main Presentation part) must be defined (physically located at /_rels/.rels in the package):

```
<Relationships xmlns="…">
  <Relationship Id="rId1"

Type="http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument"
    Target="presentation.xml"/>
</Relationships>
```

Finally, the minimum content for the Main Presentation part must be defined (physically located at /presentation.xml in the package):

```
<p:presentation xmlns:p="…">
  <p:notesSz cx="913607" cy="913607"/>
</p:presentation>
```

end example]

[*Example*: Consider a simple PresentationML document containing two slides, which both use an image as a template. Here's an example of the hierarchical folder structure that might be used for the ZIP items in the package for that document. As shown, a number of parts have their own relationship items:

```
/_rels/.rels                              Package-relationship item
/[Content_Types].xml                      Content-type item
/docProps/app.xml                         Application-Defined File Properties
part
/docProps/core.xml                        Core File Properties part
/docProps/custom.xml                      Custom File Properties part
/docProps/thumbnail.wmf                   Package thumbnail image
/ppt/presentation.xml                     Presentation part
/ppt/_rels/presentation.xml.rels          Part-relationship item
/ppt/presProps.xml                        Presentation Properties part
/ppt/tableStyles.xml                      Table Styles part
/ppt/viewProps.xml                        View Properties part
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

/ppt/handoutMasters/handoutMaster1.xml	*Handout Master part*
/ppt/handoutMasters/_rels/handoutMaster1.xml.rels	*Part-relationship item*
/ppt/media/image1.jpeg	*Slide template image*
/ppt/notesMasters/notesMaster1.xml	*Notes Master part*
/ppt/notesMasters/_rels/notesMaster1.xml.rels	*Part-relationship item*
/ppt/notesSlides/notesSlide1.xml	*Notes Slide parts*
/ppt/notesSlides/notesSlide2.xml	
/ppt/notesSlides/_rels/notesSlide1.xml.rels	*Part-relationship items*
/ppt/notesSlides/_rels/notesSlide2.xml.rels	
/ppt/slideLayouts/slideLayout1.xml	*Slide Layout parts 1–6*
/ppt/slideLayouts/slideLayout2.xml	
/ppt/slideLayouts/slideLayout3.xml	
/ppt/slideLayouts/slideLayout4.xml	
/ppt/slideLayouts/slideLayout5.xml	
/ppt/slideLayouts/slideLayout6.xml	
/ppt/slideLayouts/_rels/slideLayout1.xml.rels	*Part-relationship items*
/ppt/slideLayouts/_rels/slideLayout2.xml.rels	
/ppt/slideLayouts/_rels/slideLayout3.xml.rels	
/ppt/slideLayouts/_rels/slideLayout4.xml.rels	
/ppt/slideLayouts/_rels/slideLayout5.xml.rels	
/ppt/slideLayouts/_rels/slideLayout6.xml.rels	
/ppt/slideMasters/slideMaster1.xml	*Slide Master part*
/ppt/slideMasters/_rels/slideMaster1.xml.rels	*Part-relationship item*
/ppt/slides/slide1.xml	*Slide parts*
/ppt/slides/slide2.xml	
/ppt/slides/_rels/slide1.xml.rels	*Part-relationship items*
/ppt/slides/_rels/slide2.xml.rels	
/ppt/theme/theme1.xml	*Theme parts*
/ppt/theme/theme2.xml	
/ppt/theme/theme3.xml	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
/ppt/theme/themeOverride1.xml                    Theme Override parts
/ppt/theme/themeOverride2.xml
/ppt/theme/themeOverride3.xml
/ppt/theme/themeOverride4.xml
/ppt/theme/themeOverride5.xml
/ppt/theme/themeOverride6.xml
/ppt/theme/themeOverride7.xml
/ppt/theme/themeOverride8.xml
/ppt/theme/themeOverride9.xml
/ppt/theme/themeOverride10.xml
```

The package-relationship item contains the following:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="ppt/presentation.xml"/>
  <Relationship Id="rId3"
    Type="http://…/core-properties" Target="docProps/core.xml"/>
  <Relationship Id="rId2"
    Type="http://…/thumbnail" Target="docProps/thumbnail.wmf"/>
  <Relationship Id="rId4"
    Type="http://…/extended-properties" Target="docProps/app.xml"/>
</Relationships>
```

end example]

13.3 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to PresentationML.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

Part	Relationship Target of	Root Element	Ref.
Comment Authors	Presentation	cmAuthorLst	§13.3.1
Comments	Slide	cmLst	§13.3.2
Handout Master	Presentation	handoutMaster	§13.3.3
Notes Master	Notes Slide, Presentation	notesMaster	§13.3.4
Notes Slide	Slide	notes	§13.3.5
Presentation	PresentationML package	presentation	§13.3.6
Presentation Properties	Presentation	presentationPr	§13.3.7
Slide	Presentation	sld	§13.3.8
Slide Layout	Slide Master, Notes Slide,	sldLayout	§13.3.9

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Part	Relationship Target of	Root Element	Ref.
	Presentation, Slide, Slide Master		
Slide Master	Presentation, Slide Layout	sldMaster	§13.3.10
Slide Synchronization Data	Slide	sldSyncPr	§13.3.11
User-Defined Tags	Presentation, Slide	tagLst	§13.3.12
View Properties	Presentation	viewPr	§13.3.13

end note]

13.3.1 Comment Authors Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.commentAuthors+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/commentAuthors

An instance of this part type contains information about each author who has added a comment to the document. That information includes the author's name, initials, a unique author-ID, a last-comment-index-used count, and a display color. (The color can be used when displaying comments to distinguish comments from different authors.)

A package shall contain at most one Comment Authors part. If it exists, that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part relationship item contains a relationship to the Comment Authors part, which is stored in the ZIP item commentAuthors.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId8"
    Type="http://…/commentAuthors" Target="commentAuthors.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be cmAuthorLst.

[*Example*: Two people have authored comments in this document: Mary Smith and Peter Jones. Her initials are "mas", her author-ID is 0, and her comments' display color index is 0. Since Mary's last-comment-index-used value is 3, the next comment-index to be used for her is 4. His initials are "pjj", his author-ID is 1, and his comments' display color index is 1. Since Peter's last-comment-index-used value is 1, the next comment-index to be used for him is 2:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:cmAuthorLst xmlns:p="…" …>
  <p:cmAuthor id="0" name="Mary Smith" initials="mas" lastIdx="3"
    clrIdx="0"/>
  <p:cmAuthor id="1" name="Peter Jones" initials="pjj" lastIdx="1"
    clrIdx="1"/>
</p:cmAuthorLst>
```

end example]

A Comment Authors part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Comment Authors part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.2 Comments Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.comments+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/comments

An instance of this part type contains the comments for a single slide. Each comment is tied to its author via an author-ID. Each comment's index number and author-ID combination are unique.

A package shall contain one Comments part for each slide containing one or more comments, and each of those parts shall be the target of an implicit relationship from its corresponding Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to a Comments part, which is stored in the ZIP item ../comments/comment2.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/comments"
    Target="../comments/comment2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be cmLst .

[*Example*: The Comments part contains three comments, two created by one author, and one created by another, all at the dates and times shown. The index numbers are assigned on a per-author basis, starting at 1 for an author's first comment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:cmLst xmlns:p="…" …>
  <p:cm authorId="0" dt="2005-11-13T17:00:22.071" idx="1">
    <p:pos x="4486" y="1342"/>
    <p:text>Comment text goes here.</p:text>
  </p:cm>
  <p:cm authorId="0" dt="2005-11-13T17:00:34.849" idx="2">
    <p:pos x="3607" y="1867"/>
    <p:text>Another comment's text goes here.</p:text>
  </p:cm>
  <p:cm authorId="1" dt="2005-11-15T00:06:46.919" idx="1">
    <p:pos x="1493" y="2927"/>
    <p:text>comment …</p:text>
  </p:cm>
</p:cmLst>
```

end example]

A Comments part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Comments part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.3 Handout Master Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.handoutMaster+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/handoutMaster

An instance of this part type contains the look, position, and size of the slides, notes, header and footer text, date, or page number on the presentation's handout.

A package shall contain at most one Handout Master part, and it shall be the target of an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the Handout Master part, which is stored in the ZIP item handoutMasters/handoutMaster1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/handoutMaster"
    Target="handoutMasters/handoutMaster1.xml"/>
</Relationships>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

The root element for a part of this content type shall be handoutMaster.

[*Example*:

```
<p:handoutMaster xmlns:p="…">
  <p:cSld name="">
    …
  </p:cSld>
  <p:clrMap … />
</p:handoutMaster>
```

end example]

A Handout Master part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Handout Master part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)
- Theme (§14.2.7)
- Thumbnail (§15.2.16)

A Handout Master part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3)
- Image (§15.2.14)
- Video (§15.2.15)

A Handout Master part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

13.3.4 Notes Master Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.notesMaster+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/notesMaster

An instance of this part type contains information about the content and formatting of all notes pages.

A package shall contain at most one Notes Master part, and that part shall be the target of an implicit relationship from the Notes Slide (§13.3.5) part, as well as an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the Notes Master part, which is stored in the ZIP item notesMasters/notesMaster1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/notesMaster" Target="notesMasters/notesMaster1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be notesMaster.

[*Example*:

```
<p:notesMaster xmlns:p="…">
  <p:cSld name="">
    …
  </p:cSld>
  <p:clrMap … />
</p:notesMaster>
```

end example]

A Notes Master part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Notes Master part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Theme (§14.2.7)
- Thumbnail (§15.2.16)

A Notes Master part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3).
- Image (§15.2.14)
- Video (§15.2.15)

The Notes Master part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.5 Notes Slide Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.notesSlide+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main ain
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/notesSlide

An instance of this part type contains the notes for a single slide.

A package shall contain one Notes Slide part for each slide that contains notes. If they exist, those parts shall each be the target of an implicit relationship from the Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to a Notes Slide part, which is stored in the ZIP item .../notesSlides/notesSlide1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/notesSlide" Target="../notesSlides/notesSlide1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be notes.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

```
<p:notes xmlns:p="…">
  <p:cSld name="">
    …
  </p:cSld>
  <p:clrMapOvr>
    <a:masterClrMapping/>
  </p:clrMapOvr>
</p:notes>
```

end example]

A Notes Slide part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Notes Slide part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)
- Notes Master (§13.3.4)
- Theme Override(§14.2.8)
- Thumbnail (§15.2.16)

A Notes Slide part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3).
- Image (§15.2.14)
- Video (§15.2.15)

The Notes Slide part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.6 Presentation Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.presentation.main+xml application/vnd.openxmlformats-officedocument.presentationml.slideshow.main+xml

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	application/vnd.openxmlformats-officedocument.presentationml.template.main+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument

An instance of this part type contains the definition for a slide presentation.

A package shall contain exactly one Presentation part, and that part shall be the target of a relationship in the package-relationship item.

[*Example*: The following PresentationML's package-relationship item contains a relationship to the Presentation part, which is stored in the ZIP item ppt/presentation.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/officeDocument" Target="ppt/presentation.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be presentation.

[*Example*: This presentation contains two slides:

```
<p:presentation xmlns:p="…" … >
  <p:sldMasterIdLst>
    <p:sldMasterId
      xmlns:rel="http://…/relationships" rel:id="rId1"/>
  </p:sldMasterIdLst>
  <p:notesMasterIdLst>
    <p:notesMasterId
      xmlns:rel="http://…/relationships" rel:id="rId4"/>
  </p:notesMasterIdLst>
  <p:handoutMasterIdLst>
    <p:handoutMasterId
       xmlns:rel="http://…/relationships" rel:id="rId5"/>
  </p:handoutMasterIdLst>
  <p:sldIdLst>
    <p:sldId id="267"
      xmlns:rel="http://…/relationships" rel:id="rId2"/>
    <p:sldId id="256"
      xmlns:rel="http://…/relationships" rel:id="rId3"/>
  </p:sldIdLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <p:sldSz cx="9144000" cy="6858000"/>
    <p:notesSz cx="6858000" cy="9144000"/>
  </p:presentation>
```

end example]

A Presentation part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Presentation part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Comment Authors (§13.3.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)
- Font (§15.2.13)
- Presentation Properties (§13.3.7)
- Table Styles (§14.2.9)
- Theme (§14.2.7)
- Thumbnail (§15.2.16)
- View Properties (§13.3.13).

A Presentation part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Notes Master (§13.3.4)
- Handout Master (§13.3.3)
- Slide (§13.3.8)
- Slide Master (§13.3.10)
- User Defined Tags (§13.3.12)

13.3.7 Presentation Properties Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.presProps+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/presProps

An instance of this part type contains all the presentation's properties.

A package shall contain exactly one Presentation Properties part, and that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following Presentation part-relationship item contains a relationship to the Presentation Properties part, which is stored in the ZIP item presProps.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId6"
    Type="http://…/presProps" Target="presProps.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be presentationPr.

[*Example*:

```
<p:presentationPr xmlns:p="…" …>
  <p:clrMru>
    …
  </p:clrMru>
  …
</p:presentationPr>
```

end example]

A Presentation Properties part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Presentation Properties part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.8 Slide Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.slide+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/slide

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

A Slide part contains the contents of a single slide.

A package shall contain one Slide part per slide, and each of those parts shall be the target of an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: Consider a PresentationML document having two slides. The corresponding Presentation part-relationship item contains two relationships to Slide parts, which are stored in the ZIP items slides/slide1.xml and slides/slide2.xml:

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/slide" Target="slides/slide1.xml"/>
  <Relationship Id="rId3"
    Type="http://…/slide" Target="slides/slide2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be sld.

[*Example*: slides/slide1.xml contains:

```
<p:sld xmlns:p="…">
  <p:cSld name="">
    …
  </p:cSld>
  <p:clrMapOvr>
    …
  </p:clrMapOvr>
  <p:timing>
    <p:tnLst>
      <p:par>
        <p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot"/>
      </p:par>
    </p:tnLst>
  </p:timing>
</p:sld>
```

end example]

A Slide part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Slide part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Comments (§13.3.2)
- Custom XML Data Storage (§15.2.4)
- Notes Slide (§13.3.5)
- Theme Override (§14.2.8)
- Thumbnail (§15.2.16)
- Slide Layout (§13.3.9)
- Slide Synchronization Data (§13.3.11)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A Slide part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3).
- Image (§15.2.14)
- User Defined Tags (§13.3.12)
- Video (§15.2.15)

A Slide part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.9 Slide Layout Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.slideLayout+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/slideLayout

An instance of this part type contains the definition for a slide layout template for this presentation. This template defines the default appearance and positioning of drawing objects on this slide type when it is created.

A package shall contain one or more Slide Layout parts, and each of those parts shall be the target of an explicit relationship in the Slide Master (§13.3.10) part, as well as an implicit relationship from each of the Slide (§13.3.8) parts associated with this slide layout.

[*Example*: The following Slide Master part-relationship item contains relationships to several Slide Layout parts, which are stored in the ZIP items ../slideLayouts/slideLayout*N*.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/slideLayout"
    Target="../slideLayouts/slideLayout1.xml"/>
  <Relationship Id="rId2"
    Type="http://…/slideLayout"
    Target="../slideLayouts/slideLayout2.xml"/>
  <Relationship Id="rId3"
    Type="http://…/slideLayout"
    Target="../slideLayouts/slideLayout3.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be sldLayout.

[*Example*:

```
<p:sldLayout xmlns:p="…" matchingName="" type="title" preserve="1">
  <p:cSld name="Title Slide">
    …
  </p:cSld>
  <p:clrMapOvr>
    <a:masterClrMapping/>
  </p:clrMapOvr>
  <p:timing/>
  </p:sldLayout>
</p:sldMaster>
```

end example]

A Slide Layout part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)
- Slide Master (§13.3.10)
- Theme Override (§14.2.8)
- Thumbnail (§15.2.16)

A Slide Layout part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3).
- Image (§15.2.14)
- Video (§15.2.15)

A Slide Layout part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.10 Slide Master Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.slideMaster+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/slideMaster

An instance of this part type contains the master definition of formatting, text, and objects that appear on each slide in the presentation that is derived from this slide master.

A package shall contain one or more Slide Master parts, each of which shall be the target of an explicit relationship from the Presentation (§13.3.6) part, as well as an implicit relationship from any Slide Layout (§13.3.9) part where that slide layout is defined based on this slide master. Each can optionally be the target of a relationship in a Slide Layout (§13.3.9) part as well.

[*Example*: The following Presentation part-relationship item contains a relationship to the Slide Master part, which is stored in the ZIP item slideMasters/slideMaster1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/slideMaster" Target="slideMasters/slideMaster1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be sldMaster.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:sldMaster xmlns:p="…">
  <p:cSld name="">
    …
  </p:cSld>
  <p:clrMap … />
</p:sldMaster>
```

end example]

A Slide Master part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Slide Master part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Additional Characteristics (§15.2.1)
- Bibliography (§15.2.3)
- Custom XML Data Storage (§15.2.4)
- Theme (§14.2.7)
- Thumbnail (§15.2.16)

A Slide Master part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Audio (§15.2.2)
- Chart (§14.2.1)
- Content Part (§15.2.4)
- Diagrams: Diagram Colors(§14.2.3), Diagram Data(§14.2.4), Diagram Layout Definition(§14.2.5) and Diagram Styles (§14.2.6)
- Embedded Control Persistence (§15.2.9)
- Embedded Object (§15.2.10)
- Embedded Package (§15.2.11)
- Hyperlink (§15.3).
- Image (§15.2.14)
- Slide Layout (§13.3.9)
- Video (§15.2.15)

A Slide Master part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

13.3.11 Slide Synchronization Data Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.slideUpdateInfo+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/slideUpdateInfo

©ISO/IEC 2016 – All rights reserved

An instance of this part type contains properties specifying the current state of a slide that is being synchronized with a version of that slide stored on a central server.

A package shall contain zero or one Slide Synchronization Data part for each slide stored in the presentation, and that part shall be the target of an implicit relationship from the corresponding Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to the Slide Synchronization Data part, which is stored in the ZIP item slideUpdateInfo/slideUpdateInfo1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1" Type="http://…/slideUpdateInfo"
    Target="slideUpdateInfo/slideUpdateInfo1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be sldSyncPr.

[*Example*:

```
<p:sldSyncPr xmlns:p="…" serverSldId="1"
  serverSldModifiedTime="2006-08-12T01:31:08"
  clientInsertedTime="2006-08-12T01:34:11.227" />
```

end example]

A Slide Synchronization Data part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Slide Synchronization Data part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Slide Synchronization Server Location (§13.4)

A Slide Synchronization Data part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

13.3.12 User Defined Tags Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.tags+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/tags

©ISO/IEC 2016 – All rights reserved

An instance of this part type contains a set of user-defined properties for an object in a presentation (each property consisting of a name/value pair).

A package shall contain zero or more User Defined Tags parts, each as the target of an explicit relationship from the corresponding Presentation (§13.3.6) or Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to the User Defined Tags part, which is stored in the ZIP item tags/tag1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1" Type="http://…/tag"
    Target="tags/tag1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be tagLst.

[*Example*:

```
<p:tagLst xmlns:p="…" >
  <p:tag name="testTagName" val="testTagValue" />
</p:tagLst>
```

end example]

A User Defined Tags part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A User Defined Tags part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

13.3.13 View Properties Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.viewProps+xml
Root Namespace:	http://purl.oclc.org/ooxml/presentationml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/viewProps

An instance of this part type contains display properties for this presentation.

A package shall contain zero or one View Properties part, and if it exists, that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following Presentation part-relationship item contains a relationship to the View Properties part, which is stored in the ZIP item viewProps.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId7"
    Type="http://…/viewProps" Target="viewProps.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be viewPr.

[*Example*:

```
<p:viewPr xmlns:p="…" …>
  <p:normalViewPr showOutlineIcons="0">
    …
  </p:normalViewPr>
  <p:slideViewPr>
    …
  </p:slideViewPr>
  <p:outlineViewPr>
    …
  </p:outlineViewPr>
  <p:notesTextViewPr>
    …
  </p:notesTextViewPr>
  <p:gridSpacing cx="78028800" cy="78028800"/>
</p:viewPr>
```

end example]

A View Properties part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A View Properties part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

13.4 HTML Publish Location

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/htmlPubSaveAs

©ISO/IEC 2016 – All rights reserved

When a presentation specifies an external location to which an optional copy might be pushed in the HTML format, this relationship shall be used to target the location where the HTML copy of the presentation is published.

A package shall contain one HTML Publish Location relationship for each slide linked with an HTML publish location, and that relationships shall be an explicit relationship from the corresponding Presentation Properties (§13.3.7) part.

[*Example*: A Presentation Properties part, which stores an HTML Publish Location of http://www.openxmlformats.org/test.htm contains the following relationship in that part's relationship part:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/htmlPubSaveAs"
    Target="http://www.openxmlformats.org/test.htm" type=”External”/>
</Relationships>
```

end example]

An HTML publish location shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

13.5 Slide Synchronization Server Location

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/slideUpdateUrl

When a slide is being synchronized with a copy stored on a remote server, this relationship shall be used to target the location where the server copy of the slide is stored.

A package shall contain one Slide Synchronization Server Location relationship for each slide linked with server data, and that relationships shall be an implicit relationship from the corresponding Slide Synchronization Data (§13.3.11) part.

[*Example*: A Slide Synchronization Data part that stores information about a slide that is synchronized with a server located at http://www.openxmlformats.org/slides/ contains the following relationship in that part's relationship part item:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/slideupdateUrl"
    Target="http://www.openxmlformats.org/slides/" type=”External”/>
</Relationships>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A slide synchronization server location shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `External`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

14. DrawingML

The relationship items and parts defined in this clause are used by one or more of WordprocessingML (§11), SpreadsheetML (§12), and PresentationML (§13) environments.

14.1 Glossary of DrawingML-Specific Terms

diagram — A picture or graphical representation that is displayed using a related set of color, data, layout, and style parts. Examples of diagram types are cycle, organization chart, pyramid, target, and Venn.

14.2 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to DrawingML.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

Part	Relationship Target of	Root Element	Ref.
Chart	WordprocessingML: Main Document SpreadsheetML: Drawings PresentationML: Handout Master, Notes Master, Notes Slide, Slide Layout, Slide Master, Slide All: Chart Drawing	chartSpace	§14.2.1
Chart Drawing	All: Chart	userShapes	§14.2.2
Diagram Colors	WordprocessingML: Main Document SpreadsheetML: Drawings PresentationML: Handout Master, Notes Master, Notes Slide, Slide Layout, Slide Master, Slide	colorsDef	§14.2.3
Diagram Data	WordprocessingML: Main Document SpreadsheetML: Drawings PresentationML: Handout Master, Notes Master, Notes Slide, Slide Layout, Slide Master, Slide	dataModel	§14.2.4
Diagram Layout Definition	WordprocessingML: Main Document SpreadsheetML: Drawings PresentationML: Handout Master, Notes Master, Notes Slide, Slide Layout, Slide Master, Slide	layoutDef	§14.2.5

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Relationship Target of	Root Element	Ref.
Diagram Style	WordprocessingML: Main Document SpreadsheetML: Drawings PresentationML: Handout Master, Notes Master, Notes Slide, Slide Layout, Slide Master, Slide	styleDef	§14.2.6
Theme	WordprocessingML: Main Document SpreadsheetML: Workbook PresentationML: Handout Master, Notes Master, Presentation, Slide Master	theme	§14.2.7
Theme Override	PresentationML: Notes Slide, Slide, Slide Layout	themeOverride	§14.2.8
Table Styles	PresentationML: Presentation	tblStyleLst	§14.2.9

end note]

14.2.1 Chart Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.chart+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/chart
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/chart

An instance of this part type describes a chart.

A package shall contain a Chart part for each chart in the document. In a WordprocessingML document, each such part shall be the target of an explicit relationship in a Main Document (§11.3.10) part. In a SpreadsheetML document, each such part shall be the target of an explicit relationship in a Drawings (§12.3.8) part. In a PresentationML document, each such part shall be the target of an explicit relationship in a Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. This part is permitted to also be the target of an explicit relationship in a Chart Drawing (§14.2.2) part, if the chart that points at this Chart Drawing part is the target of a relationship from a Chartsheet part. In other words, the only time a chart can embed another chart is if the parent chart is part of a chartsheet.

[*Example*: The following Main Document part-relationship item contains relationships to two Chart parts, which are stored in the ZIP items ../charts/chart*N*.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/chart" Target="charts/chart1.xml"/>
  <Relationship Id="rId5"
    Type="http://…/chart" Target="charts/chart2.xml"/>
</Relationships>
```

The following Drawings part-relationship item contains a relationship to a Chart part, which is stored in the ZIP item ../charts/chart1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/relationships/chart" Target="../charts/chart1.xml"/>
</Relationships>
```

The following Slide part-relationship item contains relationships to two Chart parts, which are stored in the ZIP items ../charts/chart*N*.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/chart" Target="../charts/chart1.xml"/>
  <Relationship Id="rId5"
    Type="http://…/chart" Target="../charts/chart2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be chartSpace.

[*Example*: chart1.xml contains the following clustered bar chart:

```
<c:chartSpace …>
  <c:chart>
    <c:title>
      …
    </c:title>
    <c:plotArea>
      <c:layout>
        …
      </c:layout>
      <c:barChart>
        …
      </c:barChart>
    </c:plotArea>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <c:legend>
          …
        </c:legend>
      </c:chart>
      …
    </c:chartSpace>
```

end example]

For WordprocessingML and PresentationML documents, the data for a chart is not stored in the Chart part directly. Instead, it shall be stored in an embedded SpreadsheetML package (§12.2) targeted by an Embedded Package (§15.2.11) part specified by that Chart part. For SpreadsheetML documents, the data for a chart is stored directly in the Drawing's parent worksheet; no embedded SpreadsheetML package shall be used.

A Chart part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Chart part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Chart Drawing (§14.2.2)
- Embedded Package (§15.2.11)

A Chart part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

14.2.2 Chart Drawing Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.chartshapes+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/chart
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/chartUserShapes

An instance of this part type contains all basic drawing elements (shapes) which are explicitly associated with this chart. These drawing elements are automatically moved with the chart when it is moved and resized when the chart is resized.

A package is permitted to contain one Chart Drawing part per chart part, and each such part shall be the target of an explicit relationship from a Chart (§14.2.1) part.

[*Example*: The following Chart part-relationship item contains a relationship to a Chart Drawing part, which is stored in the ZIP item ../drawings/drawing1.xml:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/chartUserShapes" Target="../drawings/drawing1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be userShapes.

[*Example*:

```
<c:userShapes xmlns:cdr="…" xmlns:c="…">
  <cdr:relSizeAnchor>
    …
  </cdr:relSizeAnchor>
</c:userShapes>
```

end example]

A Chart Drawing part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Chart Drawing part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Chart (§14.2.1)

A Chart Drawing part shall not have any implicit or explicit relationships to any other part defined by ISO/IEC 29500.

14.2.3 Diagram Colors Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.diagramColors+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/diagram
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/diagramColors

An instance of this part type contains color information for a diagram.

A package shall contain exactly one Diagram Colors part per diagram. Each Diagram Colors part shall be the target of an explicit relationship in a WordprocessingML Main Document (§11.3.10), SpreadsheetML Drawings (§12.3.8), or PresentationML Slide (§13.3.8) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Colors parts, which are stored in the ZIP items ../graphics/colors*N*.xml.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
     Type="http://…/diagramColors" Target="../graphics/colors1.xml"/>
  <Relationship Id="rId8"
    Type="http://…/diagramColors" Target="../graphics/colors2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be colorsDef.

[*Example*: colors1.xml contains the following:

```
<cs:colorsDef xmlns:cs="…" uniqueId="…" minVer="12.0">
  <cs:title lang="" val="Primary Accent 2"/>
  <cs:desc lang="" val="Primary Accent 2"/>
  <cs:catLst>
    <cs:cat type="accent1" pri="11200"/>
  </cs:catLst>
  <cs:styleLbl …>
    …
  </cs:styleLbl>
  …
  <cs:styleLbl …>
    …
  </cs:styleLbl>
</cs:colorsDef>
```

end example]

A Diagram Colors part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Diagram Colors part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

14.2.4 Diagram Data Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.diagramData+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/diagram
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/diagramData

An instance of this part type contains the semantic data for a diagram.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package shall contain exactly one Diagram Data part per diagram. Each Diagram Data part shall be the target of an explicit relationship in a WordprocessingML Main Document (§11.3.10); a SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Data parts, which are stored in the ZIP items ../graphics/data*N*.xml.

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/diagramData" Target="../graphics/data1.xml"/>
  <Relationship Id="rId5"
    Type="http://…/diagramData" Target="../graphics/data2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be dataModel.

[*Example*: data1.xml contains the following:

```
<dm:dataModel xmlns:dm="…">
  <dm:ptLst>
    …
  </dm:ptLst>
  <dm:cxnLst>
    …
  </dm:cxnLst>
  <dm:bg/>
  <dm:whole/>
</dm:dataModel>
```

end example]

A Diagram Data part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Diagram Data part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

- Image (§15.2.14)

A Diagram Data part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

14.2.5 Diagram Layout Definition Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.diagramLayout+xml
Root	http://purl.oclc.org/ooxml/drawingml/diagram

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Namespace:	
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/diagramLayout

An instance of this part type is a template that describes how diagram-related data is mapped to a shape.

A package shall contain exactly one Diagram Layout Definition part per diagram. Each Layout Definition part shall be the target of an explicit relationship from a WordprocessingML Main Document (§11.3.10); a SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. If a document contains multiple diagrams having the same graphic layout definition, each of those diagrams shall have its own copy of that Diagram Layout Definition part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Layout Definition parts, which are stored in the ZIP items ../graphics/layout*N*.xml.

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/diagramLayout" Target="../graphics/layout1.xml"/>
  <Relationship Id="rId6"
    Type="http://…/diagramLayout" Target="../graphics/layout2.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be layoutDef.

[*Example*: layout1.xml contains the following:

```
<lo:layoutDef xmlns:lo="…" uniqueId="…2" minVer="12.0" defStyle="">
  <lo:title lang="" val="Hierarchy 2"/>
  <lo:desc lang="" val=""/>
  <lo:catLst>
    <lo:cat type="hierarchy" pri="2000"/>
  </lo:catLst>
  <lo:sampData>
    …
  </lo:sampData>
  <lo:styleData>
    …
  </lo:styleData>
  <lo:clrData>
    …
  </lo:clrData>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <lo:layoutNode name="Name0" styleLbl="" moveWith="">
      …
     </lo:layoutNode>
  </lo:layoutDef>
```

end example]

A Diagram Layout Definition part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Diagram Layout Definition part is permitted to have explicit relationships to the following parts and items defined by ISO/IEC 29500:

- Image (§15.2.14)

A Diagram Layout Definition part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

14.2.6 Diagram Style Part

Content Type:	application/vnd.openxmlformats-officedocument.drawingml.diagramStyle+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/diagram
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/diagramQuickStyle

An instance of this part type maps diagram semantic information to a document's theme.

A package shall contain exactly one Diagram Style part per diagram. Each Style part shall be the target of an explicit relationship from a WordprocessingML Main Document (§11.3.10); a SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Style parts, which are stored in the ZIP items ../graphics/quickStyle*N*.xml.

```
<Relationships xmlns="…">
  <Relationship Id="rId3"
    Type="http://…/diagramQuickStyle"
    Target="../graphics/quickStyle1.xml"/>
  <Relationship Id="rId7"
    Type="http://…/diagramQuickStyle"
    Target="../graphics/quickStyle2.xml"/>
</Relationships>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

The root element for a part of this content type shall be styleDef.

[*Example*: quickStyle1.xml contains the following:

```
<qs:styleDef xmlns:qs="…" uniqueId="…" minVer="12.0">
  <qs:title lang="" val="Style 2"/>
  <qs:desc lang="" val="Style 2"/>
  <qs:catLst>
    <qs:cat type="simple" pri="10200"/>
  </qs:catLst>
  <qs:scene3d>
    …
  </qs:scene3d>
  <qs:style>
    …
  </qs:style>
  <qs:styleLbl name="…">
    …
  </qs:styleLbl>
  …
  <qs:styleLbl name="…">
    …
  </qs:styleLbl>
</qs:styleDef>
```

end example]

A Diagram Style part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Diagram Style part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

14.2.7 Theme Part

Content Type:	application/vnd.openxmlformats-officedocument.theme+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/theme

An instance of this part type contains information about a document's *theme*, which is a combination of *color scheme*, *font scheme*, and *format scheme* (the latter also being referred to as *effects*). For a WordprocessingML document, the choice of theme affects the color and style of headings, among other things. For a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

SpreadsheetML document, the choice of theme affects the color and style of cell contents and charts, among other things. For a PresentationML document, the choice of theme affects the formatting of slides, handouts, and notes via the associated master, among other things.

A WordprocessingML or SpreadsheetML package shall contain zero or one Theme part, which shall be the target of an implicit relationship in a Main Document (§11.3.10) or Workbook (§12.3.23) part. A PresentationML package shall contain zero or one Theme part per Handout Master (§13.3.3), Notes Master (§13.3.4), Slide Master (§13.3.10) or Presentation (§13.3.6) part via an implicit relationship.

[*Example*: The following WordprocessingML Main Document part-relationship item contains a relationship to the Theme part, which is stored in the ZIP item theme/theme1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/theme" Target="theme/theme1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be theme.

[*Example*: theme1.xml contains the following, where the name attributes of the clrScheme, fontScheme, and fmtScheme elements correspond to the document's color scheme, font scheme, and format scheme, respectively:

```
<a:theme xmlns:a="…">
  <a:themeElements>
    <a:clrScheme name="…">
      …
    </a:clrScheme>
    <a:fontScheme name="…">
      …
    </a:fontScheme>
    <a:fmtScheme name="…">
      …
    </a:fmtScheme>
  </a:themeElements>
  <a:objectDefaults/>
</a:theme>
```

end example]

A Theme part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Theme part is permitted to contain explicit relationships to the following parts defined by ISO/IEC 29500:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Image (§15.2.14)

A Theme part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

14.2.8 Theme Override Part

Content Type:	application/vnd.openxmlformats-officedocument.themeOverride+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/themeOverride

An instance of this part type contains information about an object's *theme override*, which are overrides to the *color scheme*, *font scheme*, and *format scheme* (the latter also being referred to as *effects*) for a particular slide, notes slide, or handout.

A PresentationML package shall contain zero or one Theme Override part per Notes Slide (§13.3.5), Slide (§13.3.8), or Slide Layout (§13.3.9) part via an implicit relationship.

[*Example*: The following WordprocessingML Main Document part relationship item contains a relationship to the Theme part, which is stored in the ZIP item theme/theme1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/themeOverride" Target="theme/themeoverride1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be ossOverride.

[*Example*:

```
<a:ossOverride xmlns:a="…" >
  …
</a:ossOverride>
```

end example]

A Theme Override part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Theme Override part shall not contain implicit or explicit relationships to other parts defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

14.2.9 Table Styles Part

Content Type:	application/vnd.openxmlformats-officedocument.presentationml.tableStyles+xml
Root Namespace:	http://purl.oclc.org/ooxml/drawingml/main
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/tableStyles

An instance of this part type contains information about the table styles used by tables in this presentation. A table style defines characteristics such as row and column colors, heading row colors, and text.

A PresentationML package shall contain no more than one Table Styles part per Presentation (§13.3.6) part via an implicit relationship.

[*Example*: The following Presentation part-relationship item contains a relationship to a Table Styles part, which is stored in the ZIP item tableStyles.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/tableStyles" Target="tableStyles.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be tblStyleLst.

[*Example*: tablestyles.xml contains the following:

```
<a:tblStyleLst xmlns:a="…">
  <a:tblStyle>
    …
  </a:tblStyle>
</a:tblStyleLst>
```

end example]

A Table Styles part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Table Styles part shall not contain implicit or explicit relationships to other parts defined by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

15. Shared

The relationship items and parts defined in this clause are used by one or more of WordprocessingML (§11), SpreadsheetML (§12), and PresentationML (§13) environments.

15.1 Glossary of Shared Terms

array — An array of mathematical elements (“e”) stacked vertically in a single math zone.

build down — The process of converting mathematical text from an implementation’s professional form to an implementation’s built-down form.

build up — The process of converting mathematical text from an implementation’s built-down form to an implementation’s professional form.

built-down form — An implementation-specific linear format that may or may not include additional rich formatting in addition to another plain-text linear format (such as TeX) or the linear format defined in Unicode Technical Note 28.

built-up form — See professional form.

control — A region of active content within an Office Open XML document.

display equation — An equation that is in display mode, and thus is part of a display math zone. (Alternative names for display equation are: “display expression”, “display formula”, and “display math”.)

display mode — When mathematical text (i.e., text in one or more oMath blocks) is contained in a display math zone (i.e., an oMathPara block), the mathematical text represented in the oMath block(s) is in *display mode*.

equation array — An array of equations. See array.

inline equation — An equation that is in an inline math zone. (Alternative names for inline equation are: “inline expression”, “inline formula”, and simply “inline math”.)

instance of mathematical text — A single continuous combination of mathematical text represented by a single oMath block and the OMML elements within that oMath block.

linear format — An implementation-specific plain-text 1-dimensional representation of mathematical text.

math accent — A character that is specified as acceptable for use as an accent character by ISO/IEC 29500.

math alphanumerics — Characters with specific math styles, as defined in the Unicode Standard 5.0.

math paragraph — One or more oMath elements (instances of mathematical text) that are in display mode.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

math zone — An isolated region of text within which mathematical text is used and outside of which mathematical text is not used.

mathematical text — Any text meant to convey mathematical meaning through OMML.

n-ary operator — An operator that involves n terms when expanded. For instance, the following example uses the Unicode (ISO 10646) summation sign (U+2211) which has the official name "N-ARY SUMMATION".

$$\sum_{j=1}^{n} a_j \equiv a_1 + a_2 + \cdots + a_n$$

oMath container — An oMath block that is part of a display math zone but is not itself a math zone.

OMML — Office Math Markup Language, a Shared ML of ISO/IEC 29500.

professional form — Implementation-specific 2-D representation of mathematical text. (Also referred to as "built-up form".)

15.2 Part Summary

The subclauses subordinate to this one describe in detail each of the shared part types.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

Part	Relationship Target of	Root Element	Ref.
Additional Characteristics	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Characteristics	§15.2.1
Audio	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Not applicable	§15.2.2
Bibliography	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Sources	§15.2.3
Custom XML Data Storage	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Not applicable	§15.2.4
Custom XML Data Storage Properties	Custom XML Data Storage	datastoreItem	§15.2.6
Digital Signature Origin	WordprocessingML, SpreadsheetML, or PresentationML package	Not applicable	§15.2.7
Digital Signature XML Signature	Digital Signature Origin	Signature	§15.2.8
Embedded Control	Numerous PresentationML,	Not applicable	§15.2.9

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Relationship Target of	Root Element	Ref.
Persistence	SpreadsheetML, and WordprocessingML parts		
Embedded Object	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Not applicable	§15.2.10
Embedded Package	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Not applicable	§15.2.11
File Properties, Extended	WordprocessingML, SpreadsheetML, or PresentationML package	Properties	§ 15.2.12.3
File Properties, Core	WordprocessingML, SpreadsheetML, or PresentationML package	coreProperties	§15.2.12.1
File Properties, Custom	WordprocessingML, SpreadsheetML, or PresentationML package	properties	§15.2.12.2
Font	WordprocessingML Font Table part, PresentationML Presentation part	Not applicable	§15.2.13
Image	Numerous PresentationML, SpreadsheetML, and WordprocessingML parts	Not applicable	§15.2.14
Printer Settings	SpreadsheetML Chartsheet, Dialogsheet, Worksheet parts, WordprocessingML Main Document or Glossary Document parts	Not applicable	§15.2.15
Thumbnail	WordprocessingML, SpreadsheetML, or PresentationML package	Not applicable	§15.2.16
Video part	Numerous PresentationML and WordprocessingML parts	Not applicable	§15.2.17

end note]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

15.2.1 Additional Characteristics Part

Content Type:	application/xml
Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/additionalCharacteristics
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml

©ISO/IEC 2016 – All rights reserved

An instance of this part type contains information about additional characteristics of the producer that generated the document, when those characteristics cannot be specified using elements defined by ISO/IEC 29500. [*Note*: The contents of this part are purely informational, and do not place any requirements on subsequent consumption of the document. They are, however, intended to provide detailed information about the capabilities of the document's producer, allowing those capabilities to be factored in during subsequent processing. For example, an application which supports 100,000 spreadsheet columns might choose to limit its output to 10,000 columns when presented with a document whose characteristics indicate that it was produced by an application with that limitation, in order prevent the introduction of content which is unsupported by the original producer (as that application might be used in the future to process this document). This markup is provided by ISO/IEC 29500 in order to provide an interoperable way of storing this information. *end note*]

A package is permitted to contain zero or one Additional Characteristics parts, and each such part shall be the target of an implicit relationship from a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Presentation (§13.3.6), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to an Additional Characteristics part, which is stored in the ZIP item ../customXML/item2.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/customXmlData" Target="../customXML/item2.xml"/>
</Relationships>
```

end example]

An Additional Characteristics part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

An Additional Characteristics part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Custom XML Data Storage Properties (§15.2.6)

An Additional Characteristics part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

15.2.2 Audio Part

Content Type:	Any supported audio type. [*Note*: Some example content types are: audio/aiff — http://developer.apple.com/documentation/QuickTime/INMAC/SOUN

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	<table><tr><td></td><td>D/imsoundmgr.30.htm</td></tr><tr><td>audio/midi</td><td>http://www.midi.org/about-midi/specinfo.shtml</td></tr><tr><td>audio/ogg</td><td>http://xiph.org/vorbis/doc/Vorbis_I_spec.html</td></tr><tr><td>audio/mpeg</td><td>ISO/IEC 11172-3</td></tr></table> *end note*]
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/audio

An instance of this part type contains an audio file.

A PresentationML package is permitted to contain zero or more Sound parts, each of which shall be the target of a relationship in a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part-relationship item. [*Example*: The following Slide part-relationship item contains a relationship to a Sound part, which is stored as the file E:\Beethoven's Symphony No. 9.wma:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/audio/x-ms-wma"
    Target="file:///E:/Beethoven's%20Symphony%20No.%209.wma"
    TargetMode="External"/>
</Relationships>
```

end example]

An Audio part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

An Audio part is not stored as XML; instead, it involves a relationship target that is an audio clip.

An Audio part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

A producer that wants interoperability should use the following standard format:

- audio/mpeg ISO/IEC 11172-3

15.2.3 Bibliography Part

Content Type:	application/xml
Root Namespace:	http://purl.oclc.org/ooxml/officeDocument/bibliography

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customXml

An instance of this part type contains bibliographic data for the current package.

A package is permitted to contain zero or one Bibliography part, and each such part shall be the target of an implicit relationship in a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Bibliography part, which is stored in the ZIP item ../customXML/bib1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/customXml" Target="../customXML/bib1.xml"/>
</Relationships>
```

end example]

A Bibliography part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Bibliography part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Custom XML Data Storage Properties (§15.2.6)

A Bibliography part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

15.2.4 Content Part

Content Type:	Any supported XML content. [*Note*: Some example content types are: image/svg+xml — http://www.w3.org/TR/SVG11/ application/smil — http://www.w3.org/TR/REC-smil/ text/xml — http://www.w3.org/TR/MathML2/ *end note*] If no explicit MIME type exists for a specific XML format, text/xml shall be used. Consumers who read a value of text/xml should determine the contents by the root namespace of the contents of the part.
Root Namespace:	Various, as defined by the content type used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	[*Example*: MathML has a root namespace of http://www.w3.org/1998/Math/MathML. *end example*]
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customXml

An instance of this part type can contain XML markup of a format not defined by ISO/IEC 29500.

A package is permitted to contain zero or more Content parts, and each such part shall be the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Glossary Document (§11.3.8), Header (§11.3.9), or Main Document (§11.3.10) part in a WordprocessingML package; a Drawings (§12.3.8) part in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or a Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Content part containing SVG markup, which is stored in the ZIP item svg1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/customXml" Target="../customXML/svg1.xml"/>
</Relationships>
```

end example]

A Content part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Content part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

If a producer that wants interoperability supports equations, it should use one of the following standard formats:

- Office Open XML Math (§22.1
- W3C MathML 2.0

If a producer that wants interoperability supports ink annotations, it should use an ink annotation in this element in the following reference standard format:

- InkML http://www.w3.org/TR/inkregs

15.2.5 Custom XML Data Storage Part

Content Type:	application/xml
Root Namespace:	any XML allowed

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customXml

An instance of this part type can contain arbitrary XML. As such, an instance of this part can be used to roundtrip arbitrary custom XML data with this package.

A package is permitted to contain one or more Custom XML Data Storage parts, and each such part shall be the target of an implicit relationship in a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Presentation (§13.3.6), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Custom XML Data Storage part, which is stored in the ZIP item ../customXML/item1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/customXmlData" Target="../customXML/item1.xml"/>
</Relationships>
```

end example]

A Custom XML Data Storage part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Custom XML Data Storage part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

- Custom XML Data Storage Properties (§15.2.6)

A Custom XML Data Storage part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

15.2.6 Custom XML Data Storage Properties Part

Content Type:	application/vnd.openxmlformats-officedocument.customXmlProperties+xml
Root Namespace:	http://purl.oclc.org/ooxml/officeDocument/customXmlDataProps
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customXmlProps

An instance of this part type contains the set of properties which are specified for this custom XML data. These properties consist of a unique ID for the storage, as well as information on the set of XML schemas used by this custom XML data storage.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

A package is permitted to contain zero or more Custom XML Data Storage Properties parts, and each such part shall be the target of an implicit relationship from a Custom XML Data Storage (§15.2.4) part.

[*Example*: The following Custom XML Data Storage part-relationship item contains a relationship to a Custom XML Data Storage Properties part, which is stored in the ZIP item itemProps1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/customXmlProps" Target="itemProps1.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be datastoreItem.

[*Example*:

```
<ds:datastoreItem ds:itemID="{D85…53A}" xmlns:ds="…"/>
```

end example]

A Custom XML Data Storage Properties part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Custom XML Data Storage Properties part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.7 Digital Signature Origin Part

This part is defined in §13.2.1, “Digital Signature Origin Part”, of ISO/IEC 29500-2.

15.2.8 Digital Signature XML Signature Part

The part is defined in §13.2.2, “Digital Signature XML Signature Part”, of ISO/IEC 29500-2.

15.2.9 Embedded Control Persistence Part

Content Type:	Any supported control type. [*Note*: There are a number of possible control types. One example of a potential control type would be an Active X control, which would use the following content type: `application/vnd.ms-office.activeX+xml`. *end note*]
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/control

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

An instance of this part contains information about an embedded control in the package. This information is provided by the specified control when asked to persist. [*Example*: An application might utilize the embedded object server technology KParts or Bonobo to store an embedded object using this part. *end example*]

A package is permitted to contain one or more Embedded Control Persistence parts, and each such part shall be the target of an explicit relationship in an Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part-relationship item in a WordprocessingML package; a Worksheet part (§12.3.24) in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part-relationship item in a PresentationML package.

The content type of this part shall determine the format and contents of the embedded control.

[*Example*: The following example shows the persistence that could be used for an embedded control which is a Java applet within a WordprocessingML document (the drawing object which provides a static image representation of the control, used when the Java applet itself is unavailable, has been omitted for brevity):

```
<w:p>
  <w:r w:rsidR="005810E1">
    <w:object w:dxaOrig="1440" w:dyaOrig="1440">
      <w:drawing>
        …
      </w:drawing>
      <w:control r:id="rId5" w:name="CommandButton1" w:shapeid="1027" />
    </w:object>
  </w:r>
</w:p>
```

The relationship type for rId5 is:
`http://purl.oclc.org/ooxml/officeDocument/relationships/control`

The XML content of the part referenced by rId5 could be:

```
<applet xlink:href="../../../../Program%20Files/Application" xlink:type="simple"
xlink:show="embed" xlink:actuate="onLoad" code="CalculateApplet.class" may-
script="false"/>
```

end example]

[*Example*: The following example shows the persistence that could be used for an embedded control which is an ActiveX control within a WordprocessingML document(the drawing object which provides a static image representation of the control, used when the ActiveX control itself is unavailable, has been omitted for brevity):

```
<w:p>
  <w:r w:rsidR="005810E1">
    <w:object w:dxaOrig="1440" w:dyaOrig="1440">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <w:drawing>
          …
        </w:drawing>
        <w:control r:id="rId5" w:name="CommandButton1" w:shapeid="1027" />
      </w:object>
    </w:r>
  </w:p>
```

The relationship type for rId5 is:
`http://purl.oclc.org/ooxml/officeDocument/relationships/control`

The content type of the part referenced by rId5 could be: `application/vnd.ms-office.activeX+xml`

The XML content of the part referenced by rId5 could be:

```
<ax:ocx ax:classid="{D7053240-CE69-11CD-A777-00DD01143C57}"
    ax:persistence="persistPropertyBag"
    xmlns:ax="http://schemas.microsoft.com/office/2006/activeX">
  <ax:ocxPr ax:name="Caption" ax:value="CommandButton1" />
  <ax:ocxPr ax:name="Size" ax:value="2540;847" />
  <ax:ocxPr ax:name="FontName" ax:value="Calibri" />
  <ax:ocxPr ax:name="FontHeight" ax:value="225" />
  <ax:ocxPr ax:name="FontCharSet" ax:value="0" />
  <ax:ocxPr ax:name="FontPitchAndFamily" ax:value="2" />
  <ax:ocxPr ax:name="ParagraphAlign" ax:value="3" />
</ax:ocx>
```

end example]

An Embedded Control Persistence part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

An Embedded Control Persistence part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.10 Embedded Object Part

Content Type:	Any content type is allowed
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/oleObject

An instance of this part type can contain an embedded object produced by any embedded object server.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package is permitted to contain zero or more Embedded Object parts, and each such part shall be the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part in a WordprocessingML package; a Worksheet part (§12.3.24) in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part in a PresentationML package.

A WordprocessingML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Main Document part-relationship item. Each Embedded Object part shall have an associated image, which appears in the document as a placeholder for the corresponding embedded object.

[*Example*: Consider the case in which a WordprocessingML document has embedded in it one video object and one audio object. The following Main Document part-relationship item contains relationships to two Embedded parts (one each for the video and audio), which are stored in the ZIP items embeddings/embeddedObject*N*.bin:

```
<Relationships xmlns="…">
  <Relationship Id="rId5"
    Type="http://…/oleObject" Target="embeddings/embeddedObject1.bin"/>
  <Relationship Id="rId7"
    Type="http://…/oleObject" Target="embeddings/embeddedObject2.bin"/>
  <Relationship Id="rId4"
    Type="http://…/image" Target="media/image1.png"/>
  <Relationship Id="rId6"
    Type="http://…/image" Target="media/image2.png"/>
</Relationships>
```

example]

A SpreadsheetML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Worksheet part-relationship item.

[*Example*: Consider the case in which a SpreadsheetML document has embedded in it one video object and one audio object on one worksheet, and another audio object embedded in another worksheet. The following Worksheet Document part-relationship item contains relationships to two Embedded Object parts (one each for the video and audio), which are stored in the ZIP items ../embeddings/embeddedObject*N*.bin:

```
<Relationships xmlns="…">
   <Relationship Id="rId2"
     Type="http://…/oleObject" Target="../embeddings/embeddedObject1.bin"/>
  <Relationship Id="rId3"
    Type="http://…/oleObject" Target="../embeddings/embeddedObject2.bin"/>
</Relationships>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A PresentationML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Slide part-relationship item.

[*Example*: Consider the case in which a PresentationML document has embedded in it one video object and one audio object on one slide, and another audio object embedded on another slide. The following Slide part-relationship item contains relationships to two Embedded Object parts (one each for the video and audio), which are stored in the ZIP items ../embeddings/embeddedObject*N*.bin:

```
<Relationships xmlns="…">
  <Relationship Id="rId6"
    Type="http://…/oleObject"
    Target="../embeddings/embeddedObject1.bin"/>
  <Relationship Id="rId7"
    Type="http://…/oleObject"
    Target="../embeddings/embeddedObject2.bin"/>
</Relationships>
```

end example]

An Embedded Object part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

An Embedded Object part is permitted to have an explicit relationship to the following parts defined by ISO/IEC 29500:

- Hyperlink (§15.3)

An Embedded Object part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.11 Embedded Package Part

Content Type:	Any content type is allowed
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/package

An instance of this part type contains a complete package. For example, a WordprocessingML document might contain a SpreadsheetML or PresentationML document, in which case, the WordprocessingML document package would contain an embedded package part that defined that SpreadsheetML or PresentationML document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A package is permitted to contain zero or more Embedded Package parts, and each such part shall be the target of an explicit relationship from a Chart (§14.2.1), Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part in a WordprocessingML package; a Chart (§14.2.1), or Worksheet part (§12.3.24) in a SpreadsheetML package; or a Chart (§14.2.1), Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Presentation part-relationship item contains relationships to two Embedded Package parts: one is a SpreadsheetML package, which is stored in the ZIP item embeddings/Worksheet1.xlsx, the other is a PresentationML package, which is stored in the ZIP item embeddings/Presentation2.pptx. The image files are used as document display placeholders if the consumer cannot handle the embedded package type:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/image" Target="media/image1.emf"/>
  <Relationship Id="rId5"
    Type="http:package" Target="embeddings/Worksheet1.xlsx"/>
  <Relationship Id="rId6"
    Type="http://…/image" Target="media/image2.emf"/>
  <Relationship Id="rId7"
    Type="http://…/package" Target="embeddings/Presentation2.pptx"/>
</Relationships>
```

end example]

An Embedded Package part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

An Embedded Package part is permitted to have an explicit relationship to the following parts defined by ISO/IEC 29500:

- Hyperlink (§15.3)

An Embedded Package part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.12 File Properties

There are three kinds of file properties: *core*, *custom*, and *extended*. The *core file properties* of a package enable users to discover, get, and set common sets of properties from within that package, regardless of whether it's a WordprocessingML, SpreadsheetML, or PresentationML package. *Extended file properties* are specific to Office Open XML packages, while *custom file properties* are defined by the user, with each custom file property having a name, a value, and a type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

15.2.12.1 Core File Properties Part

This part and the related OPC part is defined in §11, “Core Properties”, of ISO/IEC 29500-2.

15.2.12.2 Custom File Properties Part

Content Type:	application/vnd.openxmlformats-officedocument.custom-properties+xml
Root Namespace:	http://purl.oclc.org/ooxml/officeDocument/customProperties
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/customProperties

An instance of this part contains the names of custom file properties that apply to the package, their values, and the types of those values. A custom file property might be the name of the client for whom the document was prepared, a date/time on which some event happened, a document number, or some Boolean status flag.

A package shall contain at most one Custom File Properties part, and that part shall be the target of a relationship in the package-relationship item for the document.

[*Example*: The following PresentationML's package-relationship item contains a relationship to a Custom File Properties part, stored in the ZIP item docProps/custom.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
  Type="http://…/custom-properties" Target="docProps/custom.xml"/>
</Relationships>
```

end example]

The root element for a part of this content type shall be Properties.

[*Example*: Here's some content markup from a WordprocessingML document, which contains four custom properties: `Client`, having a text value of "ACME Corp."; `Document number`, having a numeric value of 1543; `Recorded date`, having a date/time value of 2005-12-01; and `Special processing needed`, having a Boolean value of false:

```
<Properties … xmlns:vt="…">
  <property fmtid="{D5C…9AE}" pid="2" name="Client">
    <vt:lpwstr>ACME Corp.</vt:lpwstr>
  </property>
  <property fmtid="{D5C…9AE}" pid="3" name="Document number">
    <vt:i4>1543</vt:i4>
  </property>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <property fmtid="{D5C…9AE}" pid="4" name="Recorded date">
      <vt:filetime>2005-12-01T05:00:00Z</vt:filetime>
    </property>
    <property fmtid="{D5C…9AE}" pid="5" name="Special processing needed">
      <vt:bool>false</vt:bool>
    </property>
  </Properties>
```

end example]

A Custom File Properties part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Custom File Properties part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.12.3 Extended File Properties Part

Content Type:	application/vnd.openxmlformats-officedocument.extended-properties+xml
Root Namespace:	http://purl.oclc.org/ooxml/officeDocument/extendedProperties
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/extendedProperties

An instance of this part contains properties specific to an Office Open XML document. [*Example*: A PresentationML document specifies the number of slides in this presentation when last saved by a producer. *end example*]

A package shall contain at most one Extended File Properties part, and that part shall be the target of a relationship in the package-relationship item for the document.

[*Example*:

```
  <Relationships xmlns="…">
    <Relationship Id="rId4"
      Type="http://…/extended-properties" Target="docProps/app.xml"/>
  </Relationships>
```

end example]

The root element for a part of this content type shall be Properties.

[*Example*: Here's some content markup from a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<Properties …>
  <Template>Normal.dotm</Template>
  <TotalTime>0</TotalTime>
  <Pages>1</Pages>
  <Words>3</Words>
  <Characters>22</Characters>
  <Application>Sample Producer</Application>
  <DocSecurity>0</DocSecurity>
  <Lines>1</Lines>
  <Paragraphs>1</Paragraphs>
  …
  <AppVersion>12.0000</AppVersion>
</Properties>
```

here's some content markup from a SpreadsheetML document:

```
<Properties …>
  <Application>Sample Producer</Application>
  <HeadingPairs>
  …
  </HeadingPairs>
  <TitlesOfParts>
  …
  </TitlesOfParts>
  <Company>Consultant</Company>
  …
</Properties>
```

and here's some content markup from a PresentationML document:

```
<Properties …>
  <Template>ppt_template_sdwest05</Template>
  <TotalTime>3166</TotalTime>
  <Words>37</Words>
  <Application>Sample Producer</Application>
  <PresentationFormat>On-screen Show</PresentationFormat>
  <Paragraphs>15</Paragraphs>
  <Slides>2</Slides>
  <Notes>2</Notes>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    …
    <HeadingPairs>
    …
    </HeadingPairs>
    <TitlesOfParts>
    …
    </TitlesOfParts>
    …
  </Properties>
```

end example]

A Extended File Properties part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

An Extended File Properties part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

15.2.13 Font Part

Content Type:	application/x-fontdata application/x-font-ttf application/vnd.openxmlformats-officedocument.obfuscatedFont
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/font

An instance of this part type contains a given font embedded directly into the document. (This is useful when using custom fonts or fonts that are not widely distributed.)

Fonts stored in a Font part can be stored in one of the following formats, identified by the associated content type:

- `application/x-fontdata` specifies that the font shall be stored in the Embedded OpenType Format of http://www.w3.org/Submission/2008/SUBM-EOT-20080305
- `application/x-font-ttf` specifies that the font shall be stored in a format conforming to Open Font Structure defined in ISO/IEC 14496-22:2008 §3.5. [*Note*: The TrueType Collection format defined in ISO/IEC 14496-22:2008 §3.6 cannot be used. *end note*]
- application/vnd.openxmlformats officedocument.obfuscatedFont specifies that the font is obfuscated using the algorithm specified by Font Embedding (§17.8.1). The source font shall be stored in a format conforming to Open Font Structure defined in ISO/IEC 14496-22:2008 §3.5. [*Note*: The TrueType Collection format defined in ISO/IEC 14496-22:2008 §3.6 cannot be used. *end note*] Only packages of type WordprocessingML are permitted to reference this content type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If a font is stored in the ISO/IEC 14496-22:2007 format, it shall only be used when stored as an individual font. [*Note:* Font collections should be converted into individual fonts before they are embedded using this part. *end note*]

A package shall contain zero or more Font parts, and for each that exists, that part shall be the target of an explicit relationship in the Font Table (§11.3.5), or Presentation (§13.3.6) part.

A Font part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Font part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.14 Image Part

<table>
<tr><td>Content Type:</td><td>Any supported image type.

[Note: Some example content types are:
<table>
<tr><td>image/gif</td><td>http://www.w3.org/Graphics/GIF/spec-gif89a.txt</td></tr>
<tr><td>image/png</td><td>ISO/IEC 15948:2003 http://www.libpng.org/pub/png/spec/</td></tr>
<tr><td>image/tiff</td><td>http://partners.adobe.com/public/developer/tiff/index.html#spec</td></tr>
<tr><td>image/pict</td><td>http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html</td></tr>
<tr><td>image/jpeg</td><td>http://www.w3.org/Graphics/JPEG/</td></tr>
</table>
end note]</td></tr>
<tr><td>Root Namespace:</td><td>Not applicable</td></tr>
<tr><td>Source Relationship:</td><td>http://purl.oclc.org/ooxml/officeDocument/relationships/image</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An image can be stored in a package as a ZIP item. Image ZIP items shall be identified by an image part relationship and the appropriate content type.

A package is permitted to contain zero or more Image parts, and each such part shall be the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), Drawing (§12.3.8), or Main Document (§11.3.10) part in a WordprocessingML package or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10 part in a PresentationML package.

[*Example*: The following PresentationML's package-relationship item contains one relationship, for the slide template jpeg image stored in the ZIP item ../media/image1.jpeg:

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId8"
    Type="http://…/image" Target="../media/image1.jpeg"/>
</Relationships>
```

end example]

An Image part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

An Image part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

A producer that wants interoperability should use one of the following standard formats:

- image/png ISO/IEC 15948:2003, http://www.libpng.org/pub/png/spec/
- image/jpeg, http://www.w3.org/Graphics/JPEG

15.2.15 Printer Settings Part

Content Type:	application/vnd.openxmlformats-officedocument.spreadsheetml.printerSettings (in SpreadsheetML documents) application/vnd.openxmlformats-officedocument.wordprocessingml.printerSettings (in WordprocessingML documents) application/vnd.openxmlformats-officedocument.presentationml.printerSettings (in PresentationML documents)
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An instance of this part type contains information about the initialization and environment of a printer or a display device. The layout of this information is application-defined.

[*Note*: It is recommended that a Printer Settings Part contain well documented XML content for improved interoperability; however, there is no requirement on the format of the content contained in a Printer Settings Part. *end note*]

[*Example*: An Office Open XML producer on Windows might store the DEVMODE structure defined here: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/prntspol_8nle.asp, while an application on the Mac OS might choose to store the print record defined here: http://developer.apple.com/documentation/Printing/index.html. *end example*]

©ISO/IEC 2016 – All rights reserved

A SpreadsheetML package is permitted to contain at most one Printer Settings part per Chartsheet, Dialogsheet, or Worksheet part, and that part shall be the target of an implicit relationship from a Chartsheet (§12.3.2), Dialogsheet (§12.3.7), or Worksheet (§12.3.24) part. A WordprocessingML package is permitted to contain zero or more Printer Settings parts, one per sectPr element, each a target of an explicit relationship from a Main Document (§11.3.10) or Glossary Document (§11.3.8) part. A PresentationML package is permitted to contain at most one Printer Settings part, and that part shall be the target of an implicit relationship from a Presentation (§13.3.6) part.

[*Example*: The following SpreadsheetML Worksheet part-relationship item contains a relationship to a Printer Settings part, which is stored in the ZIP item ../printerSettings/printerSettings1.xml:

```
<Relationships xmlns="…">
  <Relationship Id="rId4"
    Type="http://…/printerSettings"
    Target="../printerSettings/printerSettings1.xml"/>
</Relationships>
```

where the contents of PrinterSettings1.xml contains the following XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PrinterSettings xmlns="…">
    <PrinterSetting name="PropertyName" value="PropertyValue" />
</PrinterSettings>
```

end example]

A Printer Settings part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Printer Settings part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

15.2.16 Thumbnail Part

<table>
<tr><td>Content Type:</td><td>Any supported image type.

[Note: Some example content types are:
<table>
<tr><td>image/gif</td><td>http://www.w3.org/Graphics/GIF/spec-gif89a.txt</td></tr>
<tr><td>image/png</td><td>ISO/IEC 15948:2003 http://www.libpng.org/pub/png/spec/</td></tr>
<tr><td>image/tiff</td><td>http://partners.adobe.com/public/developer/tiff/index.html#spec</td></tr>
<tr><td>image/pict</td><td>http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html</td></tr>
<tr><td>image/jpeg</td><td>http://www.w3.org/Graphics/JPEG/</td></tr>
</table>
end note]</td></tr>
<tr><td>Root Namespace:</td><td>Not applicable</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/metadata/thumbnail

To help end-users identify parts of a package or the package as a whole, images, called *thumbnails*, can be stored in that package. Each thumbnail image is generated by the package producer and is stored in the package as a ZIP item. There are no limitations on the size or dimensions of the thumbnail produced, and applications are free to scale the images as desired.

Thumbnail ZIP items shall be identified by either a package-relationship item or a part-relationship item. Packages shall not contain more than one thumbnail relationship associated with the package as a whole, or more than one thumbnail relationship per package part.

[*Example*: The following PresentationML's package-relationship item contains one relationship, for the metafile image stored in the ZIP item thumbnail.wmf:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/thumbnail" Target="docProps/thumbnail.wmf"/>
</Relationships>
```

end example]

A Thumbnail part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be `Internal`).

A Thumbnail part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

15.2.17 Video Part

Content Type:	Any supported video type. [*Note*: Some example content types are: video/avi — http://www.the-labs.com/Video/odmlff2-avidef.pdf video/mpg — ISO/IEC 13818 video/mpeg — ISO/IEC 13818 video/ogg — http://www.theora.org/doc/Theora.pdf video/quicktime — http://developer.apple.com/documentation/QuickTime/ video/vc1 — http://tools.ietf.org/html/rfc4425 *end note*]
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/video

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

An instance of this part type contains a video file.

A PresentationML package is permitted to contain zero or more Video parts, each of which shall be the target of an explicit relationship in a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. A WordprocessingML package is permitted to contain zero or more Video parts, each of which shall be the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part.

[*Example*: The following Slide part-relationship item contains a relationship to a Video part, which is stored as the file E:\Video demo.avi:

```
<Relationships xmlns="…">
  <Relationship Id="rId2"
    Type="http://…/video"
    Target="file:///E:\Video%20demo.avi" TargetMode="External"/>
</Relationships>
```

end example]

A Video part is not stored as XML; instead, it involves a relationship target that is a video clip.

A Video part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

A Video part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

A producer that wants interoperability should use the following standard format:

- video/mpeg ISO/IEC 13818

15.3 Hyperlinks

Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink

A hyperlink can be stored in a package as a relationship. Hyperlinks shall be identified by containing a target which specifies the destination of the given hyperlink.

[*Example*: The following WordprocessingML Footnote part's relationship part contains one relationship, for the hyperlink http://schemas.openxmlformats.org/wordprocessingml/:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationships xmlns="…">
  <Relationship Id="rId1"
    Type="http://…/hyperlink"
    Target="http://schemas.openxmlformats.org/wordprocessingml/"
    TargetMode="External"/>
</Relationships>
```

end example]

A hyperlink target can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be `Internal` or `External`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

16. Part Overview

This clause is informative.

For convenience, the following subclauses specify the root elements for each part and, when applicable, link to the appropriate subclause that defines the root element for that part within the package. [*Note*: The terms part and package, as used here, are defined in ISO/IEC 29500-2. *end note*] Note that the contents of some of these parts are not defined in this Part of ISO/IEC 29500, in which case, the root element is listed as "Not applicable" and the Reference is "n/a". For those parts where the Root Element is listed as "Not Applicable", a definition of the part's use and how its content shall be structured is defined in this Part 1 of ISO/IEC 29500.

16.1 WordprocessingML Summary

Part	Root Element	Ref.
Alternative Format Import	Not applicable	n/a
Comments	comments	§17.13.4.6
Document Settings	settings	§17.15.1.78
Endnotes	endnotes	§17.11.8
Font Table	fonts	§17.8.3.11
Footer	ftr	§17.10.3
Footnotes	footnotes	§17.11.15
Glossary Document	glossaryDocument	§17.12.10
Header	hdr	§17.10.4
Mail Merge Recipient Data	recipients	§17.14.29
Main Document	document	§17.2.3
Numbering Definitions	numbering	§17.9.16
Style Definitions	styles	§17.7.4.18
Web Settings	webSettings	§17.15.2.45

16.2 SpreadsheetML Summary

Part	Root Element	Ref.
Calculation Chain	calcChain	§18.6.2
Chartsheet	chartsheet	§18.3.1.12
Comments	comments	§18.7.6
Connections	connections	§18.13.2
Custom Property	Not applicable	n/a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Root Element	Ref.
Custom XML Mappings	MapInfo	§18.16.3
Dialogsheet	dialogsheet	§18.3.1.34
Drawing	wsDr	§20.5.2.35
External Workbook References	externalLink	§18.14.8
Metadata	metadata	§18.9.8
Pivot Table	pivotTableDefinition	§18.10.1.73
Pivot Table Cache Definition	pivotCacheDefinition	§18.10.1.67
Pivot Table Cache Records	pivotCacheRecords	§18.10.1.68
Query Table	queryTable	§18.12.2
Shared String Table	sst	§18.4.9
Shared Workbook Revision Headers	headers	§18.11.1.1
Shared Workbook Revision Log	revisions	§18.11.1.16
Shared Workbook User Data	users	§18.11.2.2
Single Cell Table Definitions	singleXmlCells	§18.5.2.2
Styles	styleSheet	§18.8.39
Table Definition	table	§18.5.1.2
Volatile Dependencies	volTypes	§18.15.6
Workbook	workbook	§18.2.27
Worksheet	worksheet	§18.3.1.99

16.3 PresentationML Summary

Part	Root Element	Ref.
Comment Authors	cmAuthorLst	§19.4.3
Comments	cmLst	§19.4.4
Handout Master	handoutMaster	§19.3.1.24
Notes Master	notesMaster	§19.3.1.27
Notes Slide	notes	§19.3.1.26
Presentation	presentation	§19.2.1.26
Presentation Properties	presentationPr	§19.2.1.27
Slide	sld	§19.3.1.38
Slide Layout	sldLayout	§19.3.1.39
Slide Master	sldMaster	§19.3.1.42
Slide Synchronization Data	sldSyncPr	§19.6.1
User-Defined Tags	tagLst	§19.3.3.2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Root Element	Ref.
View Properties	viewPr	§19.2.2.18

16.4 DrawingML Summary

Part	Root Element	Ref.
Chart	chartSpace	§21.2.2.29
Chart Drawing	userShapes	§21.2.2.220
Diagram Colors	colorsDef	§21.4.4.3
Diagram Data	dataModel	§21.4.2.10
Diagram Layout Definition	layoutDef	§21.4.2.16
Diagram Style	styleDef	§21.4.5.7
Theme	theme	§20.1.6.9
Theme Override	themeOverride	§20.1.6.12
Table Styles	tblStyleLst	§20.1.4.2.27

16.5 Shared Summary

Part	Root Element	Ref.
Additional Characteristics	additionalCharacteristics	§22.7.2.1
Audio	Not applicable	n/a
Bibliography	Sources	§22.6.2.60
Custom XML Data Storage	Not applicable	n/a
Custom XML Data Storage Properties	datastoreItem	§22.5.2.1
Digital Signature Origin	Not applicable	n/a
Digital Signature XML Signature	Signature	Defined in ISO/IEC 29500-2
Embedded Control Persistence	Not applicable	n/a
Embedded Object	Not applicable	n/a
Embedded Package	Not applicable	n/a
File Properties, Core	coreProperties	Defined in ISO/IEC 29500-2
File Properties, Custom	Properties	§22.3.2.1
File Properties, Extended	Properties	§22.2.2.21
Font	Not applicable	n/a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Part	Root Element	Ref.
Image	Not applicable	n/a
Printer Settings	Not applicable	n/a
Thumbnail	Not applicable	n/a
Video	Not applicable	n/a

End of informative text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17. WordprocessingML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, “Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)”. *end note*]

The subordinate subclauses specify the semantics for the XML markup comprising a WordprocessingML document, as defined by §11 of this Part of ISO/IEC 29500.

17.1 Table of Contents

This subclause is informative.

17.2 Main Document Story 189
17.2.1 background (Document Background) 189
17.2.2 body (Document Body) 194
17.2.3 document (Document) 195

17.3 Paragraphs and Rich Formatting 195
17.3.1 Paragraphs 196
17.3.1.1 adjustRightInd (Automatically Adjust Right Indent When Using Document Grid) 197
17.3.1.2 autoSpaceDE (Automatically Adjust Spacing of Latin and East Asian Text) 197
17.3.1.3 autoSpaceDN (Automatically Adjust Spacing of East Asian Text and Numbers) 198
17.3.1.4 bar (Paragraph Border Between Facing Pages) 198
17.3.1.5 between (Paragraph Border Between Identical Paragraphs) 199
17.3.1.6 bidi (Right to Left Paragraph Layout) 201
17.3.1.7 bottom (Paragraph Border Below Identical Paragraphs) 201
17.3.1.8 cnfStyle (Paragraph Conditional Formatting) 203
17.3.1.9 contextualSpacing (Ignore Spacing Above and Below When Using Identical Styles) 208
17.3.1.10 divId (Associated HTML div ID) 209
17.3.1.11 framePr (Text Frame Properties) 210
17.3.1.12 ind (Paragraph Indentation) 221
17.3.1.13 jc (Paragraph Alignment) 226
17.3.1.14 keepLines (Keep All Lines On One Page) 227
17.3.1.15 keepNext (Keep Paragraph With Next Paragraph) 228
17.3.1.16 kinsoku (Use East Asian Typography Rules for First and Last Character per Line) 230
17.3.1.17 left (Left Paragraph Border) 233
17.3.1.18 mirrorIndents (Use Left/Right Indents as Inside/Outside Indents) 234
17.3.1.19 numPr (Numbering Definition Instance Reference) 235
17.3.1.20 outlineLvl (Associated Outline Level) 235
17.3.1.21 overflowPunct (Allow Punctuation to Extend Past Text Extents) 236
17.3.1.22 p (Paragraph) 237
17.3.1.23 pageBreakBefore (Start Paragraph on Next Page) 239
17.3.1.24 pBdr (Paragraph Borders) 241
17.3.1.25 pPr (Previous Paragraph Properties) 241
17.3.1.26 pPr (Paragraph Properties) 242

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.3.1.27 pStyle (Referenced Paragraph Style) ... 242
17.3.1.28 right (Right Paragraph Border) ... 244
17.3.1.29 rPr (Run Properties for the Paragraph Mark) ... 245
17.3.1.30 rPr (Previous Run Properties for the Paragraph Mark) ... 246
17.3.1.31 shd (Paragraph Shading) ... 247
17.3.1.32 snapToGrid (Use Document Grid Settings for Inter-Line Paragraph Spacing) ... 247
17.3.1.33 spacing (Spacing Between Lines and Above/Below Paragraph) ... 248
17.3.1.34 suppressAutoHyphens (Suppress Hyphenation for Paragraph) ... 253
17.3.1.35 suppressLineNumbers (Suppress Line Numbers for Paragraph) ... 254
17.3.1.36 suppressOverlap (Prevent Text Frames From Overlapping) ... 255
17.3.1.37 tab (Custom Tab Stop) ... 255
17.3.1.38 tabs (Set of Custom Tab Stops) ... 257
17.3.1.39 textAlignment (Vertical Character Alignment on Line) ... 257
17.3.1.40 textboxTightWrap (Allow Surrounding Paragraphs to Tight Wrap to Text Box Contents) ... 259
17.3.1.41 textDirection (Paragraph Text Flow Direction) ... 260
17.3.1.42 top (Paragraph Border Above Identical Paragraphs) ... 261
17.3.1.43 topLinePunct (Compress Punctuation at Start of a Line) ... 263
17.3.1.44 widowControl (Allow First/Last Line to Display on a Separate Page) ... 263
17.3.1.45 wordWrap (Allow Line Breaking At Character Level) ... 264
17.3.2 Run ... 265
17.3.2.1 b (Bold) ... 266
17.3.2.2 bCs (Complex Script Bold) ... 266
17.3.2.3 bdo (Bidirectional Override) ... 267
17.3.2.4 bdr (Text Border) ... 268
17.3.2.5 caps (Display All Characters As Capital Letters) ... 268
17.3.2.6 color (Run Content Color) ... 269
17.3.2.7 cs (Use Complex Script Formatting on Run) ... 272
17.3.2.8 dir (Bidirectional Embedding Level) ... 273
17.3.2.9 dstrike (Double Strikethrough) ... 275
17.3.2.10 eastAsianLayout (East Asian Typography Settings) ... 275
17.3.2.11 effect (Animated Text Effect) ... 279
17.3.2.12 em (Emphasis Mark) ... 280
17.3.2.13 emboss (Embossing) ... 281
17.3.2.14 fitText (Manual Run Width) ... 281
17.3.2.15 highlight (Text Highlighting) ... 283
17.3.2.16 i (Italics) ... 284
17.3.2.17 iCs (Complex Script Italics) ... 284
17.3.2.18 imprint (Imprinting) ... 285
17.3.2.19 kern (Font Kerning) ... 285
17.3.2.20 lang (Languages for Run Content) ... 287
17.3.2.21 noProof (Do Not Check Spelling or Grammar) ... 289
17.3.2.22 oMath (Office Open XML Math) ... 289
17.3.2.23 outline (Display Character Outline) ... 290
17.3.2.24 position (Vertically Raised or Lowered Text) ... 291
17.3.2.25 r (Text Run) ... 292
17.3.2.26 rFonts (Run Fonts) ... 293
17.3.2.27 rPr (Previous Run Properties) ... 306
17.3.2.28 rPr (Run Properties) ... 307

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.2.29 rStyle (Referenced Character Style) 308
17.3.2.30 rtl (Right To Left Text) 309
17.3.2.31 shadow (Shadow) 311
17.3.2.32 shd (Run Shading) 311
17.3.2.33 smallCaps (Small Caps) 312
17.3.2.34 snapToGrid (Use Document Grid Settings For Inter-Character Spacing) 312
17.3.2.35 spacing (Character Spacing Adjustment) 313
17.3.2.36 specVanish (Paragraph Mark Is Always Hidden) 314
17.3.2.37 strike (Single Strikethrough) 315
17.3.2.38 sz (Non-Complex Script Font Size) 315
17.3.2.39 szCs (Complex Script Font Size) 316
17.3.2.40 u (Underline) 317
17.3.2.41 vanish (Hidden Text) 321
17.3.2.42 vertAlign (Subscript/Superscript Text) 322
17.3.2.43 w (Expanded/Compressed Text) 323
17.3.2.44 webHidden (Web Hidden Text) 324
17.3.3 Run Content 324
17.3.3.1 br (Break) 325
17.3.3.2 contentPart (Content Part) 326
17.3.3.3 control (Embedded Control) 329
17.3.3.4 cr (Carriage Return) 331
17.3.3.5 dayLong (Date Block - Long Day Format) 332
17.3.3.6 dayShort (Date Block - Short Day Format) 332
17.3.3.7 delText (Deleted Text) 333
17.3.3.8 dirty (Invalidated Field Cache) 334
17.3.3.9 drawing (DrawingML Object) 334
17.3.3.10 hps (Phonetic Guide Text Font Size) 334
17.3.3.11 hpsBaseText (Phonetic Guide Base Text Font Size) 335
17.3.3.12 hpsRaise (Distance Between Phonetic Guide Text and Phonetic Guide Base Text) 336
17.3.3.13 lastRenderedPageBreak (Position of Last Calculated Page Break) 337
17.3.3.14 lid (Language ID for Phonetic Guide) 337
17.3.3.15 monthLong (Date Block - Long Month Format) 338
17.3.3.16 monthShort (Date Block - Short Month Format) 339
17.3.3.17 movie (Embedded Video) 339
17.3.3.18 noBreakHyphen (Non Breaking Hyphen Character) 341
17.3.3.19 object (Embedded Object) 342
17.3.3.20 objectEmbed (Embedded Object Properties) 343
17.3.3.21 objectLink (Linked Object Properties) 345
17.3.3.22 pgNum (Page Number Block) 348
17.3.3.23 ptab (Absolute Position Tab Character) 348
17.3.3.24 rt (Phonetic Guide Text) 350
17.3.3.25 ruby (Phonetic Guide) 351
17.3.3.26 rubyAlign (Phonetic Guide Text Alignment) 352
17.3.3.27 rubyBase (Phonetic Guide Base Text) 352
17.3.3.28 rubyPr (Phonetic Guide Properties) 353
17.3.3.29 softHyphen (Optional Hyphen Character) 354
17.3.3.30 sym (Symbol Character) 355
17.3.3.31 t (Text) 356

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.3.3.32 tab (Tab Character) ... 357
17.3.3.33 yearLong (Date Block - Long Year Format) ... 358
17.3.3.34 yearShort (Date Block - Short Year Format) ... 358
17.3.4 Border Properties (CT_Border) ... 359
17.3.5 Shading Properties (CT_Shd) ... 366

17.4 Tables ... 373

17.4.1 bidiVisual (Visually Right to Left Table) ... 374
17.4.2 bottom (Table Cell Bottom Margin Exception) ... 376
17.4.3 bottom (Table Cell Bottom Border) ... 376
17.4.4 bottom (Table Bottom Border) ... 377
17.4.5 bottom (Table Cell Bottom Margin Default) ... 378
17.4.6 cantSplit (Table Row Cannot Break Across Pages) ... 379
17.4.7 cnfStyle (Table Row Conditional Formatting) ... 380
17.4.8 cnfStyle (Table Cell Conditional Formatting) ... 386
17.4.9 divId (Associated HTML div ID) ... 391
17.4.10 end (Table Cell Trailing Margin Exception) ... 393
17.4.11 end (Table Cell Trailing Margin Default) ... 394
17.4.12 end (Table Cell Trailing Edge Border) ... 395
17.4.13 end (Table Trailing Edge Border) ... 396
17.4.14 gridAfter (Grid Columns After Last Cell) ... 396
17.4.15 gridBefore (Grid Columns Before First Cell) ... 398
17.4.16 gridCol (Grid Column Definition) ... 399
17.4.17 gridSpan (Grid Columns Spanned by Current Table Cell) ... 400
17.4.18 header (Header Cell Reference) ... 402
17.4.19 headers (Header Cells Associated With Table Cell) ... 404
17.4.20 hidden (Hidden Table Row Marker) ... 406
17.4.21 hideMark (Ignore End Of Cell Marker In Row Height Calculation) ... 407
17.4.22 insideH (Table Inside Horizontal Edges Border) ... 408
17.4.23 insideH (Table Cell Inside Horizontal Edges Border) ... 409
17.4.24 insideV (Table Inside Vertical Edges Border) ... 410
17.4.25 insideV (Table Cell Inside Vertical Edges Border) ... 410
17.4.26 jc (Table Alignment Exception) ... 411
17.4.27 jc (Table Row Alignment) ... 412
17.4.28 jc (Table Alignment) ... 413
17.4.29 noWrap (Don't Wrap Cell Content) ... 414
17.4.30 shd (Table Shading Exception) ... 416
17.4.31 shd (Table Shading) ... 417
17.4.32 shd (Table Cell Shading) ... 418
17.4.33 start (Table Cell Leading Edge Border) ... 418
17.4.34 start (Table Cell Leading Margin Default) ... 419
17.4.35 start (Table Cell Leading Margin Exception) ... 420
17.4.36 start (Table Leading Edge Border) ... 421
17.4.37 tbl (Table) ... 421
17.4.38 tblBorders (Table Borders) ... 422
17.4.39 tblBorders (Table Borders Exceptions) ... 424
17.4.40 tblCaption (Table Caption) ... 425
17.4.41 tblCellMar (Table Cell Margin Exceptions) ... 426

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.4.42 tblCellMar (Table Cell Margin Defaults) 427
17.4.43 tblCellSpacing (Table Row Cell Spacing) 428
17.4.44 tblCellSpacing (Table Cell Spacing Exception) 429
17.4.45 tblCellSpacing (Table Cell Spacing Default) 430
17.4.46 tblDescription (Table Description) 431
17.4.47 tblGrid (Previous Table Grid) 432
17.4.48 tblGrid (Table Grid) 433
17.4.49 tblHeader (Repeat Table Row on Every New Page) 434
17.4.50 tblInd (Table Indent from Leading Margin) 435
17.4.51 tblInd (Table Indent from Leading Margin Exception) 436
17.4.52 tblLayout (Table Layout) 437
17.4.53 tblLayout (Table Layout Exception) 438
17.4.54 tblLook (Table Style Conditional Formatting Settings Exception) 439
17.4.55 tblLook (Table Style Conditional Formatting Settings) 440
17.4.56 tblOverlap (Floating Table Allows Other Tables to Overlap) 441
17.4.57 tblpPr (Floating Table Positioning) 443
17.4.58 tblPr (Previous Table Properties) 449
17.4.59 tblPr (Table Properties) 450
17.4.60 tblPrEx (Table-Level Property Exceptions) 451
17.4.61 tblPrEx (Previous Table-Level Property Exceptions) 453
17.4.62 tblStyle (Referenced Table Style) 455
17.4.63 tblW (Preferred Table Width) 456
17.4.64 tblW (Preferred Table Width Exception) 457
17.4.65 tc (Table Cell) 458
17.4.66 tcBorders (Table Cell Borders) 460
17.4.67 tcFitText (Fit Text Within Cell) 463
17.4.68 tcMar (Single Table Cell Margins) 463
17.4.69 tcPr (Table Cell Properties) 464
17.4.70 tcPr (Previous Table Cell Properties) 465
17.4.71 tcW (Preferred Table Cell Width) 466
17.4.72 textDirection (Table Cell Text Flow Direction) 468
17.4.73 tl2br (Table Cell Top Left to Bottom Right Diagonal Border) 469
17.4.74 top (Table Cell Top Border) 469
17.4.75 top (Table Cell Top Margin Default) 470
17.4.76 top (Table Top Border) 471
17.4.77 top (Table Cell Top Margin Exception) 472
17.4.78 tr (Table Row) 472
17.4.79 tr2bl (Table Cell Top Right to Bottom Left Diagonal Border) 474
17.4.80 trHeight (Table Row Height) 475
17.4.81 trPr (Table Row Properties) 477
17.4.82 trPr (Previous Table Row Properties) 478
17.4.83 vAlign (Table Cell Vertical Alignment) 479
17.4.84 vMerge (Vertically Merged Cell) 480
17.4.85 wAfter (Preferred Width After Table Row) 482
17.4.86 wBefore (Preferred Width Before Table Row) 483
17.4.87 Table Measurement (CT_TblWidth) 484

17.5 Custom Markup 485

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.5.1 Custom XML and Smart Tags 485
17.5.1.1 attr (Custom XML Attribute) 488
17.5.1.2 attr (Smart Tag Property) 489
17.5.1.3 customXml (Inline-Level Custom XML Element) 491
17.5.1.4 customXml (Cell-Level Custom XML Element) 492
17.5.1.5 customXml (Row-Level Custom XML Element) 494
17.5.1.6 customXml (Block-Level Custom XML Element) 495
17.5.1.7 customXmlPr (Custom XML Element Properties) 496
17.5.1.8 placeholder (Custom XML Element Placeholder Text) 497
17.5.1.9 smartTag (Inline-Level Smart Tag) 498
17.5.1.10 smartTagPr (Smart Tag Properties) 500
17.5.2 Structured Document Tags 500
17.5.2.1 alias (Friendly Name) 501
17.5.2.2 bibliography (Bibliography Structured Document Tag) 502
17.5.2.3 calendar (Date Picker Calendar Type) 503
17.5.2.4 citation (Citation Structured Document Tag) 504
17.5.2.5 comboBox (Combo Box Structured Document Tag) 504
17.5.2.6 dataBinding (XML Mapping) 506
17.5.2.7 date (Date Structured Document Tag) 509
17.5.2.8 dateFormat (Date Display Mask) 510
17.5.2.9 docPart (Document Part Reference) 512
17.5.2.10 docPartCategory (Document Part Category Filter) 513
17.5.2.11 docPartGallery (Document Part Gallery Filter) 514
17.5.2.12 docPartList (Document Part Gallery Structured Document Tag) 515
17.5.2.13 docPartObj (Built-In Document Part Structured Document Tag) 516
17.5.2.14 docPartUnique (Built-In Document Part) 517
17.5.2.15 dropDownList (Drop-Down List Structured Document Tag) 517
17.5.2.16 equation (Equation Structured Document Tag) 519
17.5.2.17 group (Group Structured Document Tag) 520
17.5.2.18 id (Unique ID) 520
17.5.2.19 label (Structured Document Tag Label) 521
17.5.2.20 lid (Date Picker Language ID) 523
17.5.2.21 listItem (Combo Box List Item) 523
17.5.2.22 listItem (Drop-Down List Item) 525
17.5.2.23 lock (Locking Setting) 527
17.5.2.24 picture (Picture Structured Document Tag) 529
17.5.2.25 placeholder (Structured Document Tag Placeholder Text) 529
17.5.2.26 richText (Rich Text Structured Document Tag) 530
17.5.2.27 rPr (Run Properties For Structured Document Tag Contents) 530
17.5.2.28 rPr (Structured Document Tag End Character Run Properties) 532
17.5.2.29 sdt (Block-Level Structured Document Tag) 532
17.5.2.30 sdt (Row-Level Structured Document Tag) 533
17.5.2.31 sdt (Inline-Level Structured Document Tag) 534
17.5.2.32 sdt (Cell-Level Structured Document Tag) 535
17.5.2.33 sdtContent (Cell-Level Structured Document Tag Content) 536
17.5.2.34 sdtContent (Block-Level Structured Document Tag Content) 537
17.5.2.35 sdtContent (Row-Level Structured Document Tag Content) 538
17.5.2.36 sdtContent (Inline-Level Structured Document Tag Content) 539

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.5.2.37 sdtEndPr (Structured Document Tag End Character Properties) 540
17.5.2.38 sdtPr (Structured Document Tag Properties) 540
17.5.2.39 showingPlcHdr (Current Contents Are Placeholder Text) 540
17.5.2.40 storeMappedDataAs (Custom XML Data Date Storage Format) 541
17.5.2.41 tabIndex (Structured Document Tag Navigation Order Index) 543
17.5.2.42 tag (Programmatic Tag) 544
17.5.2.43 temporary (Remove Structured Document Tag When Contents Are Edited) 545
17.5.2.44 text (Plain Text Structured Document Tag) 546

17.6 Sections 547
17.6.1 bidi (Right to Left Section Layout) 548
17.6.2 bottom (Bottom Border) 549
17.6.3 col (Single Column Definition) 558
17.6.4 cols (Column Definitions) 559
17.6.5 docGrid (Document Grid) 561
17.6.6 formProt (Only Allow Editing of Form Fields) 566
17.6.7 left (Left Border) 566
17.6.8 lnNumType (Line Numbering Settings) 574
17.6.9 paperSrc (Paper Source Information) 576
17.6.10 pgBorders (Page Borders) 577
17.6.11 pgMar (Page Margins) 580
17.6.12 pgNumType (Page Numbering Settings) 584
17.6.13 pgSz (Page Size) 586
17.6.14 printerSettings (Reference to Printer Settings Data) 588
17.6.15 right (Right Border) 589
17.6.16 rtlGutter (Gutter on Right Side of Page) 596
17.6.17 sectPr (Document Final Section Properties) 598
17.6.18 sectPr (Section Properties) 599
17.6.19 sectPr (Previous Section Properties) 601
17.6.20 textDirection (Text Flow Direction) 603
17.6.21 top (Top Border) 603
17.6.22 type (Section Type) 612
17.6.23 vAlign (Vertical Text Alignment on Page) 613

17.7 Styles 614
17.7.1 Style Inheritance 614
17.7.2 Style Hierarchy 615
17.7.3 Toggle Properties 616
17.7.4 General Style Properties 618
17.7.4.1 aliases (Alternate Style Names) 619
17.7.4.2 autoRedefine (Automatically Merge User Formatting Into Style Definition) 620
17.7.4.3 basedOn (Parent Style ID) 622
17.7.4.4 hidden (Hide Style From User Interface) 624
17.7.4.5 latentStyles (Latent Style Information) 625
17.7.4.6 link (Linked Style Reference) 629
17.7.4.7 locked (Style Cannot Be Applied) 630
17.7.4.8 lsdException (Latent Style Exception) 631
17.7.4.9 name (Primary Style Name) 634
17.7.4.10 next (Style For Next Paragraph) 635

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.7.4.11 personal (E-Mail Message Text Style) ... 637
17.7.4.12 personalCompose (E-Mail Message Composition Style) ... 637
17.7.4.13 personalReply (E-Mail Message Reply Style) ... 638
17.7.4.14 qFormat (Primary Style) ... 638
17.7.4.15 rsid (Revision Identifier for Style Definition) ... 639
17.7.4.16 semiHidden (Hide Style From Main User Interface) ... 640
17.7.4.17 style (Style Definition) ... 641
17.7.4.18 styles (Style Definitions) ... 646
17.7.4.19 uiPriority (Optional User Interface Sorting Order) ... 646
17.7.4.20 unhideWhenUsed (Remove Semi-Hidden Property When Style Is Used) ... 647
17.7.5 Document Defaults ... 648
17.7.5.1 docDefaults (Document Default Paragraph and Run Properties) ... 648
17.7.5.2 pPr (Paragraph Properties) ... 649
17.7.5.3 pPrDefault (Default Paragraph Properties) ... 650
17.7.5.4 rPr (Run Properties) ... 650
17.7.5.5 rPrDefault (Default Run Properties) ... 651
17.7.6 Table Styles ... 652
17.7.6.1 pPr (Table Style Conditional Formatting Paragraph Properties) ... 655
17.7.6.2 rPr (Table Style Conditional Formatting Run Properties) ... 656
17.7.6.3 tblPr (Table Style Conditional Formatting Table Properties) ... 657
17.7.6.4 tblPr (Style Table Properties) ... 657
17.7.6.5 tblStyleColBandSize (Number of Columns in Column Band) ... 658
17.7.6.6 tblStylePr (Style Conditional Table Formatting Properties) ... 659
17.7.6.7 tblStyleRowBandSize (Number of Rows in Row Band) ... 661
17.7.6.8 tcPr (Table Style Conditional Formatting Table Cell Properties) ... 662
17.7.6.9 tcPr (Style Table Cell Properties) ... 663
17.7.6.10 trPr (Table Style Conditional Formatting Table Row Properties) ... 663
17.7.6.11 trPr (Style Table Row Properties) ... 664
17.7.7 Numbering Styles ... 664
17.7.8 Paragraph Styles ... 665
17.7.8.1 Numbering in Paragraph Styles ... 666
17.7.8.2 pPr (Style Paragraph Properties) ... 668
17.7.9 Run (Character) Styles ... 669
17.7.9.1 rPr (Run Properties) ... 670

17.8 Fonts ... 670

17.8.1 Font Embedding ... 671
17.8.2 Font Substitution ... 672
17.8.3 Elements ... 672
17.8.3.1 altName (Alternate Names for Font) ... 672
17.8.3.2 charset (Character Set Supported By Font) ... 674
17.8.3.3 embedBold (Bold Style Font Style Embedding) ... 674
17.8.3.4 embedBoldItalic (Bold Italic Font Style Embedding) ... 677
17.8.3.5 embedItalic (Italic Font Style Embedding) ... 679
17.8.3.6 embedRegular (Regular Font Style Embedding) ... 681
17.8.3.7 embedSystemFonts (Embed Common System Fonts) ... 683
17.8.3.8 embedTrueTypeFonts (Embed TrueType Fonts) ... 683
17.8.3.9 family (Font Family) ... 683

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.8.3.10 font (Properties for a Single Font) 684
17.8.3.11 fonts (Font Table Root Element) 685
17.8.3.12 notTrueType (Not a TrueType outline Font) 686
17.8.3.13 panose1 (Panose-1 Typeface Classification Number) 686
17.8.3.14 pitch (Font Pitch) 687
17.8.3.15 saveSubsetFonts (Subset Fonts When Embedding) 688
17.8.3.16 sig (Supported Unicode Subranges and Code Pages) 688

17.9 Numbering 692

17.9.1 abstractNum (Abstract Numbering Definition) 694
17.9.2 abstractNumId (Abstract Numbering Definition Reference) 695
17.9.3 ilvl (Numbering Level Reference) 697
17.9.4 isLgl (Display All Levels Using Arabic Numerals) 698
17.9.5 lvl (Numbering Level Override Definition) 699
17.9.6 lvl (Numbering Level Definition) 702
17.9.7 lvlJc (Justification) 705
17.9.8 lvlOverride (Numbering Level Definition Override) 706
17.9.9 lvlPicBulletId (Picture Numbering Symbol Definition Reference) 708
17.9.10 lvlRestart (Restart Numbering Level Symbol) 709
17.9.11 lvlText (Numbering Level Text) 712
17.9.12 multiLevelType (Abstract Numbering Definition Type) 713
17.9.13 name (Abstract Numbering Definition Name) 714
17.9.14 nsid (Abstract Numbering Definition Identifier) 715
17.9.15 num (Numbering Definition Instance) 716
17.9.16 numbering (Numbering Definitions) 719
17.9.17 numFmt (Numbering Format) 719
17.9.18 numId (Numbering Definition Instance Reference) 720
17.9.19 numIdMacAtCleanup (Last Reviewed Abstract Numbering Definition) 721
17.9.20 numPicBullet (Picture Numbering Symbol Definition) 722
17.9.21 numStyleLink (Numbering Style Reference) 723
17.9.22 pPr (Numbering Level Associated Paragraph Properties) 725
17.9.23 pStyle (Paragraph Style's Associated Numbering Level) 725
17.9.24 rPr (Numbering Symbol Run Properties) 727
17.9.25 start (Starting Value) 728
17.9.26 startOverride (Numbering Level Starting Value Override) 729
17.9.27 styleLink (Numbering Style Definition) 731
17.9.28 suff (Content Between Numbering Symbol and Paragraph Text) 732
17.9.29 tmpl (Numbering Template Code) 733

17.10 Headers and Footers 734

17.10.1 evenAndOddHeaders (Different Even/Odd Page Headers and Footers) 735
17.10.2 footerReference (Footer Reference) 736
17.10.3 ftr (Footer) 739
17.10.4 hdr (Header) 741
17.10.5 headerReference (Header Reference) 743
17.10.6 titlePg (Different First Page Headers and Footers) 746

17.11 Footnotes and Endnotes 747

17.11.1 continuationSeparator (Continuation Separator Mark) 749

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.11.2 endnote (Endnote Content) ... 750
17.11.3 endnote (Special Endnote List) ... 752
17.11.4 endnotePr (Document-Wide Endnote Properties) ... 753
17.11.5 endnotePr (Section-Wide Endnote Properties) ... 754
17.11.6 endnoteRef (Endnote Reference Mark) ... 755
17.11.7 endnoteReference (Endnote Reference) ... 757
17.11.8 endnotes (Document Endnotes) ... 759
17.11.9 footnote (Special Footnote List) ... 760
17.11.10 footnote (Footnote Content) ... 761
17.11.11 footnotePr (Section-Wide Footnote Properties) ... 763
17.11.12 footnotePr (Document-Wide Footnote Properties) ... 764
17.11.13 footnoteRef (Footnote Reference Mark) ... 765
17.11.14 footnoteReference (Footnote Reference) ... 767
17.11.15 footnotes (Document Footnotes) ... 769
17.11.16 noEndnote (Suppress Endnotes In Document) ... 770
17.11.17 numFmt (Endnote Numbering Format) ... 770
17.11.18 numFmt (Footnote Numbering Format) ... 772
17.11.19 numRestart (Footnote and Endnote Numbering Restart Location) ... 773
17.11.20 numStart (Footnote and Endnote Numbering Starting Value) ... 774
17.11.21 pos (Footnote Placement) ... 776
17.11.22 pos (Endnote Placement) ... 778
17.11.23 separator (Footnote/Endnote Separator Mark) ... 779

17.12 Glossary Document ... 780

17.12.1 behavior (Entry Insertion Behavior) ... 781
17.12.2 behaviors (Entry Insertion Behaviors) ... 784
17.12.3 category (Entry Categorization) ... 784
17.12.4 description (Description for Entry) ... 785
17.12.5 docPart (Glossary Document Entry) ... 786
17.12.6 docPartBody (Contents of Glossary Document Entry) ... 787
17.12.7 docPartPr (Glossary Document Entry Properties) ... 789
17.12.8 docParts (List of Glossary Document Entries) ... 789
17.12.9 gallery (Gallery Associated With Entry) ... 790
17.12.10 glossaryDocument (Glossary Document Root Element) ... 791
17.12.11 guid (Entry ID) ... 792
17.12.12 name (Category Associated With Entry) ... 792
17.12.13 name (Entry Name) ... 794
17.12.14 style (Associated Paragraph Style Name) ... 795
17.12.15 type (Entry Type) ... 796
17.12.16 types (Entry Types) ... 797

17.13 Annotations ... 798

17.13.1 Inline Annotations ... 798
17.13.2 "Cross Structure" Annotations ... 799
17.13.3 Property Annotations ... 800
17.13.4 Comments ... 801
17.13.4.1 annotationRef (Comment Information Block) ... 803
17.13.4.2 comment (Comment Content) ... 803
17.13.4.3 commentRangeEnd (Comment Anchor Range End) ... 806

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.13.4.4 commentRangeStart (Comment Anchor Range Start) 808
17.13.4.5 commentReference (Comment Content Reference Mark) 810
17.13.4.6 comments (Comments Collection) 812
17.13.5 Revisions 812
17.13.5.1 cellDel (Table Cell Deletion) 814
17.13.5.2 cellIns (Table Cell Insertion) 816
17.13.5.3 cellMerge (Vertically Merged/Split Table Cells) 819
17.13.5.4 customXmlDelRangeEnd (Custom XML Markup Deletion End) 823
17.13.5.5 customXmlDelRangeStart (Custom XML Markup Deletion Start) 825
17.13.5.6 customXmlInsRangeEnd (Custom XML Markup Insertion End) 827
17.13.5.7 customXmlInsRangeStart (Custom XML Markup Insertion Start) 829
17.13.5.8 customXmlMoveFromRangeEnd (Custom XML Markup Move Source End) 832
17.13.5.9 customXmlMoveFromRangeStart (Custom XML Markup Move Source Start) 834
17.13.5.10 customXmlMoveToRangeEnd (Custom XML Markup Move Destination Location End) 837
17.13.5.11 customXmlMoveToRangeStart (Custom XML Markup Move Destination Location Start) 839
17.13.5.12 del (Deleted Table Row) 842
17.13.5.13 del (Deleted Math Control Character) 844
17.13.5.14 del (Deleted Run Content) 845
17.13.5.15 del (Deleted Paragraph) 847
17.13.5.16 ins (Inserted Math Control Character) 850
17.13.5.17 ins (Inserted Table Row) 851
17.13.5.18 ins (Inserted Run Content) 853
17.13.5.19 ins (Inserted Numbering Properties) 855
17.13.5.20 ins (Inserted Paragraph) 858
17.13.5.21 moveFrom (Move Source Paragraph) 860
17.13.5.22 moveFrom (Move Source Run Content) 862
17.13.5.23 moveFromRangeEnd (Move Source Location Container - End) 864
17.13.5.24 moveFromRangeStart (Move Source Location Container - Start) 868
17.13.5.25 moveTo (Move Destination Run Content) 875
17.13.5.26 moveTo (Move Destination Paragraph) 878
17.13.5.27 moveToRangeEnd (Move Destination Location Container - End) 880
17.13.5.28 moveToRangeStart (Move Destination Location Container - Start) 883
17.13.5.29 pPrChange (Revision Information for Paragraph Properties) 890
17.13.5.30 rPrChange (Revision Information for Run Properties on the Paragraph Mark) 892
17.13.5.31 rPrChange (Revision Information for Run Properties) 894
17.13.5.32 sectPrChange (Revision Information for Section Properties) 896
17.13.5.33 tblGridChange (Revision Information for Table Grid Column Definitions) 898
17.13.5.34 tblPrChange (Revision Information for Table Properties) 899
17.13.5.35 tblPrExChange (Revision Information for Table-Level Property Exceptions) 901
17.13.5.36 tcPrChange (Revision Information for Table Cell Properties) 903
17.13.5.37 trPrChange (Revision Information for Table Row Properties) 905
17.13.6 Bookmarks 907
17.13.6.1 bookmarkEnd (Bookmark End) 908
17.13.6.2 bookmarkStart (Bookmark Start) 910
17.13.7 Range Permissions 917
17.13.7.1 permEnd (Range Permission End) 918
17.13.7.2 permStart (Range Permission Start) 920
17.13.8 Spelling and Grammar 927

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.13.8.1 proofErr (Proofing Error Anchor) ... 928

17.14 Mail Merge ... 929

17.14.1 active (Record Is Included in Mail Merge) ... 931
17.14.2 activeRecord (Record Currently Displayed In Merged Document) ... 932
17.14.3 addressFieldName (Column Containing E-mail Address) ... 934
17.14.4 checkErrors (Mail Merge Error Reporting Setting) ... 935
17.14.5 colDelim (Column Delimiter for Data Source) ... 936
17.14.6 column (Index of Column Being Mapped) ... 937
17.14.7 column (Index of Column Containing Unique Values for Record) ... 939
17.14.8 connectString (Data Source Connection String) ... 940
17.14.9 dataSource (Data Source File Path) ... 941
17.14.10 dataType (Data Source Type) ... 943
17.14.11 destination (Merged Document Destination) ... 944
17.14.12 doNotSuppressBlankLines (Remove Blank Lines from Merged Documents) ... 945
17.14.13 dynamicAddress (Use Country-Based Address Field Ordering) ... 945
17.14.14 fHdr (First Row of Data Source Contains Column Names) ... 946
17.14.15 fieldMapData (External Data Source to Merge Field Mapping) ... 947
17.14.16 headerSource (Header Definition File Path) ... 947
17.14.17 lid (Merge Field Name Language ID) ... 949
17.14.18 linkToQuery (Query Contains Link to External Query File) ... 950
17.14.19 mailAsAttachment (Merged Document To E-Mail Attachment) ... 950
17.14.20 mailMerge (Mail Merge Settings) ... 951
17.14.21 mailSubject (Merged E-mail or Fax Subject Line) ... 952
17.14.22 mainDocumentType (Source Document Type) ... 953
17.14.23 mappedName (Predefined Merge Field Name) ... 954
17.14.24 name (Data Source Name for Column) ... 956
17.14.25 odso (Office Data Source Object Settings) ... 957
17.14.26 query (Query For Data Source Records To Merge) ... 958
17.14.27 recipientData (Data About Single Data Source Record) ... 959
17.14.28 recipientData (Reference to Inclusion/Exclusion Data for Data Source) ... 960
17.14.29 recipients (Inclusion/Exclusion Data for Data Source) ... 962
17.14.30 src (ODSO Data Source File Path) ... 963
17.14.31 table (Data Source Table Name) ... 964
17.14.32 type (ODSO Data Source Type) ... 966
17.14.33 type (Merge Field Mapping) ... 966
17.14.34 udl (UDL Connection String) ... 967
17.14.35 uniqueTag (Unique Value for Record) ... 969
17.14.36 viewMergedData (View Merged Data Within Document) ... 969

17.15 Settings ... 970

17.15.1 Document Settings ... 971
17.15.1.1 activeWritingStyle (Grammar Checking Settings) ... 971
17.15.1.2 alignBordersAndEdges (Align Paragraph and Table Borders with Page Border) ... 973
17.15.1.3 alwaysMergeEmptyNamespace (Do Not Mark Custom XML Elements With No Namespace As Invalid) 974
17.15.1.4 alwaysShowPlaceholderText (Use Custom XML Element Names as Default Placeholder Text) 975
17.15.1.5 attachedSchema (Attached Custom XML Schema) ... 976
17.15.1.6 attachedTemplate (Attached Document Template) ... 977

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.7 autoCaption (Single Automatic Captioning Setting) 978
17.15.1.8 autoCaptions (Automatic Captioning Settings) 980
17.15.1.9 autoFormatOverride (Allow Automatic Formatting to Override Formatting Protection Settings) 981
17.15.1.10 autoHyphenation (Automatically Hyphenate Document Contents When Displayed) 981
17.15.1.11 bookFoldPrinting (Book Fold Printing) 982
17.15.1.12 bookFoldPrintingSheets (Number of Pages Per Booklet) 984
17.15.1.13 bookFoldRevPrinting (Reverse Book Fold Printing) 986
17.15.1.14 bordersDoNotSurroundFooter (Page Border Excludes Footer) 988
17.15.1.15 bordersDoNotSurroundHeader (Page Border Excludes Header) 988
17.15.1.16 caption (Single Caption Type Definition) 989
17.15.1.17 captions (Caption Settings) 998
17.15.1.18 characterSpacingControl (Character-Level Whitespace Compression) 1001
17.15.1.19 clickAndTypeStyle (Paragraph Style Applied to Automatically Generated Paragraphs) 1001
17.15.1.20 clrSchemeMapping (Theme Color Mappings) 1003
17.15.1.21 compat (Compatibility Settings) 1008
17.15.1.22 consecutiveHyphenLimit (Maximum Number of Consecutively Hyphenated Lines) 1008
17.15.1.23 decimalSymbol (Radix Point for Field Code Evaluation) 1009
17.15.1.24 defaultTableStyle (Default Table Style for Newly Inserted Tables) 1011
17.15.1.25 defaultTabStop (Distance Between Automatic Tab Stops) 1012
17.15.1.26 displayBackgroundShape (Display Background Objects When Displaying Document) 1013
17.15.1.27 displayHorizontalDrawingGridEvery (Distance between Horizontal Gridlines) 1014
17.15.1.28 displayVerticalDrawingGridEvery (Distance between Vertical Gridlines) 1015
17.15.1.29 documentProtection (Document Editing Restrictions) 1017
17.15.1.30 documentType (Document Classification) 1022
17.15.1.31 docVar (Single Document Variable) 1023
17.15.1.32 docVars (Document Variables) 1024
17.15.1.33 doNotAutoCompressPictures (Do Not Automatically Compress Images) 1024
17.15.1.34 doNotDemarcateInvalidXml (Do Not Show Visual Indicator For Invalid Custom XML Markup) 1025
17.15.1.35 doNotDisplayPageBoundaries (Do Not Display Visual Boundary For Header/Footer or Between Pages) 1025
17.15.1.36 doNotEmbedSmartTags (Remove Smart Tags When Saving) 1027
17.15.1.37 doNotHyphenateCaps (Do Not Hyphenate Words in ALL CAPITAL LETTERS) 1027
17.15.1.38 doNotIncludeSubdocsInStats (Do Not Include Content in Text Boxes, Footnotes, and Endnotes in Document Statistics) 1028
17.15.1.39 doNotShadeFormData (Do Not Show Visual Indicator For Form Fields) 1029
17.15.1.40 doNotTrackFormatting (Do Not Track Formatting Revisions When Tracking Revisions) 1029
17.15.1.41 doNotTrackMoves (Do Not Use Move Syntax When Tracking Revisions) 1031
17.15.1.42 doNotUseMarginsForDrawingGridOrigin (Do Not Use Margins for Drawing Grid Origin) 1032
17.15.1.43 doNotValidateAgainstSchema (Do Not Validate Custom XML Markup Against Schemas) 1032
17.15.1.44 drawingGridHorizontalOrigin (Drawing Grid Horizontal Origin Point) 1032
17.15.1.45 drawingGridHorizontalSpacing (Drawing Grid Horizontal Grid Unit Size) 1034
17.15.1.46 drawingGridVerticalOrigin (Drawing Grid Vertical Origin Point) 1035
17.15.1.47 drawingGridVerticalSpacing (Drawing Grid Vertical Grid Unit Size) 1036
17.15.1.48 forceUpgrade (Upgrade Document on Open) 1038
17.15.1.49 formsDesign (Structured Document Tag Placeholder Text Should be Resaved) 1038
17.15.1.50 gutterAtTop (Position Gutter At Top of Page) 1039

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.1.51 hideGrammaticalErrors (Do Not Display Visual Indication of Grammatical Errors) 1040
17.15.1.52 hideSpellingErrors (Do Not Display Visual Indication of Spelling Errors) 1040
17.15.1.53 hyphenationZone (Hyphenation Zone) 1040
17.15.1.54 ignoreMixedContent (Ignore Mixed Content When Validating Custom XML Markup) 1042
17.15.1.55 linkStyles (Automatically Update Styles From Document Template) 1043
17.15.1.56 listSeparator (List Separator for Field Code Evaluation) 1044
17.15.1.57 mirrorMargins (Mirror Page Margins) 1045
17.15.1.58 noLineBreaksAfter (Custom Set of Characters Which Cannot End a Line) 1046
17.15.1.59 noLineBreaksBefore (Custom Set Of Characters Which Cannot Begin A Line) 1047
17.15.1.60 noPunctuationKerning (Never Kern Punctuation Characters) 1049
17.15.1.61 printFormsData (Only Print Form Field Content) 1049
17.15.1.62 printFractionalCharacterWidth (Print Fractional Character Widths) 1051
17.15.1.63 printPostScriptOverText (Print PostScript Codes With Document Text) 1051
17.15.1.64 printTwoOnOne (Print Two Pages Per Sheet) 1051
17.15.1.65 proofState (Spelling and Grammatical Checking State) 1053
17.15.1.66 readModeInkLockDown (Freeze Document Layout) 1055
17.15.1.67 removeDateAndTime (Remove Date and Time from Annotations) 1057
17.15.1.68 removePersonalInformation (Remove Personal Information from Document Properties) 1058
17.15.1.69 revisionView (Visibility of Annotation Types) 1058
17.15.1.70 rsid (Single Session Revision Save ID) 1060
17.15.1.71 rsidRoot (Original Document Revision Save ID) 1061
17.15.1.72 rsids (Listing of All Revision Save ID Values) 1062
17.15.1.73 saveFormsData (Only Save Form Field Content) 1063
17.15.1.74 saveInvalidXml (Allow Saving Document As XML File When Custom XML Markup Is Invalid) 1064
17.15.1.75 savePreviewPicture (Generate Thumbnail For Document On Save) 1064
17.15.1.76 saveThroughXslt (Custom XSL Transform To Use When Saving As XML File) 1065
17.15.1.77 saveXmlDataOnly (Only Save Custom XML Markup) 1066
17.15.1.78 settings (Document Settings) 1067
17.15.1.79 showEnvelope (Show E-Mail Message Header) 1068
17.15.1.80 showXMLTags (Show Visual Indicators for Custom XML Markup Start/End Locations) 1068
17.15.1.81 smartTagType (Supplementary Smart Tag Information) 1068
17.15.1.82 strictFirstAndLastChars (Use Strict Kinsoku Rules for Japanese Text) 1070
17.15.1.83 styleLockQFSet (Prevent Replacement of Styles Part) 1070
17.15.1.84 styleLockTheme (Prevent Modification of Themes Part) 1071
17.15.1.85 stylePaneFormatFilter (Suggested Filtering for List of Document Styles) 1071
17.15.1.86 stylePaneSortMethod (Suggested Sorting for List of Document Styles) 1073
17.15.1.87 summaryLength (Percentage of Document to Use When Generating Summary) 1074
17.15.1.88 themeFontLang (Theme Font Languages) 1074
17.15.1.89 trackRevisions (Track Revisions to Document) 1077
17.15.1.90 updateFields (Automatically Recalculate Fields on Open) 1079
17.15.1.91 useXSLTWhenSaving (Save Document as XML File through Custom XSL Transform) 1080
17.15.1.92 view (Document View Setting) 1080
17.15.1.93 writeProtection (Write Protection) 1081
17.15.1.94 zoom (Magnification Setting) 1085
17.15.2 Web Page Settings 1087
17.15.2.1 allowPNG (Allow PNG as Graphic Format) 1087
17.15.2.2 blockQuote (Data for HTML blockquote Element) 1088

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.2.3 bodyDiv (Data for HTML body Element) 1089
17.15.2.4 bottom (Bottom Border for HTML div) 1090
17.15.2.5 color (Frameset Splitter Color) 1091
17.15.2.6 div (Information About Single HTML div Element) 1094
17.15.2.7 divBdr (Set of Borders for HTML div) 1097
17.15.2.8 divs (Information about HTML div Elements) 1098
17.15.2.9 divsChild (Child div Elements Contained within Current div) 1100
17.15.2.10 doNotOrganizeInFolder (Do Not Place Supporting Files in Subdirectory) 1102
17.15.2.11 doNotRelyOnCSS (Do Not Rely on CSS for Font Face Formatting) 1102
17.15.2.12 doNotSaveAsSingleFile (Recommend Web Page Format over Single File Web Page Format) 1103
17.15.2.13 doNotUseLongFileNames (Do Not Use File Names Longer than 8.3 Characters) 1104
17.15.2.14 encoding (Output Encoding When Saving as Web Page) 1104
17.15.2.15 flatBorders (Frameset Splitter Border Style) 1105
17.15.2.16 frame (Single Frame Properties) 1106
17.15.2.17 frameLayout (Frameset Layout) 1108
17.15.2.18 frameset (Nested Frameset Definition) 1109
17.15.2.19 frameset (Root Frameset Definition) 1111
17.15.2.20 framesetSplitbar (Frameset Splitter Properties) 1112
17.15.2.21 left (Left Border for HTML div) 1113
17.15.2.22 linkedToFile (Maintain Link to Existing File) 1114
17.15.2.23 longDesc (Frame Long Description) 1115
17.15.2.24 marBottom (Bottom Margin for HTML div) 1118
17.15.2.25 marH (Top and Bottom Margin for Frame) 1119
17.15.2.26 marLeft (Left Margin for HTML div) 1121
17.15.2.27 marRight (Right Margin for HTML div) 1122
17.15.2.28 marTop (Top Margin for HTML div) 1124
17.15.2.29 marW (Left and Right Margin for Frame) 1126
17.15.2.30 name (Frame Name) 1127
17.15.2.31 noBorder (Do Not Display Frameset Splitters) 1129
17.15.2.32 noResizeAllowed (Frame Cannot Be Resized) 1131
17.15.2.33 optimizeForBrowser (Disable Features Not Supported by Target Web Profile) 1132
17.15.2.34 pixelsPerInch (Pixels per Inch for Graphics/Images) 1133
17.15.2.35 right (Right Border for HTML div) 1134
17.15.2.36 saveSmartTagsAsXml (Save Smart Tag Data in XML Property Bag) 1135
17.15.2.37 scrollbar (Scrollbar Display Option) 1136
17.15.2.38 sourceFileName (Source File for Frame) 1138
17.15.2.39 sz (Frame Size) 1140
17.15.2.40 sz (Nested Frameset Size) 1142
17.15.2.41 targetScreenSz (Target Screen Size for Web Page) 1144
17.15.2.42 title (Frame or Frameset Title) 1144
17.15.2.43 top (Top Border for HTML div) 1146
17.15.2.44 w (Frameset Splitter Width) 1147
17.15.2.45 webSettings (Web Page Settings) 1149
17.15.3 Language Compatibility Settings 1149
17.15.3.1 adjustLineHeightInTable (Add Document Grid Line Pitch To Lines in Table Cells) 1150
17.15.3.2 applyBreakingRules (Use Legacy Ethiopic and Amharic Line Breaking Rules) 1151
17.15.3.3 balanceSingleByteDoubleByteWidth (Balance Single Byte and Double Byte Characters) 1152

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.3.4 compatSetting (Custom Compatibility Setting) 1153
17.15.3.5 doNotExpandShiftReturn (Don't Justify Lines Ending in Soft Line Break) 1155
17.15.3.6 doNotLeaveBackslashAlone (Display Backslash As Yen Sign) 1155
17.15.3.7 spaceForUL (Add Additional Space Below Baseline For Underlined East Asian Text) 1156
17.15.3.8 ulTrailSpace (Underline All Trailing Spaces) 1157

17.16 Fields and Hyperlinks 1158
17.16.1 Syntax 1159
17.16.2 XML representation 1166
17.16.3 Formulas and expressions 1168
17.16.3.1 Constants 1169
17.16.3.2 Bookmarks 1169
17.16.3.3 Operators 1169
17.16.3.4 Functions 1170
17.16.3.5 Table cell references 1171
17.16.4 Field formatting 1172
17.16.4.1 Date and time formatting 1172
17.16.4.2 Numeric formatting 1184
17.16.4.3 General formatting 1185
17.16.5 Field definitions 1195
17.16.5.1 ADDRESSBLOCK 1197
17.16.5.2 ADVANCE 1198
17.16.5.3 ASK 1200
17.16.5.4 AUTHOR 1201
17.16.5.5 AUTOTEXT 1202
17.16.5.6 AUTOTEXTLIST 1202
17.16.5.7 BIBLIOGRAPHY 1203
17.16.5.8 CITATION 1204
17.16.5.9 COMMENTS 1205
17.16.5.10 COMPARE 1206
17.16.5.11 CREATEDATE 1207
17.16.5.12 DATABASE 1207
17.16.5.13 DATE 1209
17.16.5.14 DOCPROPERTY 1210
17.16.5.15 DOCVARIABLE 1212
17.16.5.16 EDITTIME 1212
17.16.5.17 FILENAME 1213
17.16.5.18 FILESIZE 1213
17.16.5.19 FILLIN 1214
17.16.5.20 FORMCHECKBOX 1215
17.16.5.21 FORMDROPDOWN 1215
17.16.5.22 FORMTEXT 1215
17.16.5.23 GOTOBUTTON 1216
17.16.5.24 GREETINGLINE 1217
17.16.5.25 HYPERLINK 1217
17.16.5.26 IF 1218
17.16.5.27 INCLUDEPICTURE 1219
17.16.5.28 INCLUDETEXT 1219

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.5.29 INDEX 1221
17.16.5.30 KEYWORDS 1223
17.16.5.31 LASTSAVEDBY 1223
17.16.5.32 LINK 1224
17.16.5.33 LISTNUM 1225
17.16.5.34 MACROBUTTON 1229
17.16.5.35 MERGEFIELD 1230
17.16.5.36 MERGEREC 1230
17.16.5.37 MERGESEQ 1231
17.16.5.38 NEXT 1231
17.16.5.39 NEXTIF 1232
17.16.5.40 NOTEREF 1232
17.16.5.41 NUMCHARS 1233
17.16.5.42 NUMPAGES 1234
17.16.5.43 NUMWORDS 1234
17.16.5.44 PAGE 1235
17.16.5.45 PAGEREF 1235
17.16.5.46 PRINT 1236
17.16.5.47 PRINTDATE 1236
17.16.5.48 PRIVATE 1237
17.16.5.49 QUOTE 1237
17.16.5.50 RD 1238
17.16.5.51 REF 1239
17.16.5.52 REVNUM 1240
17.16.5.53 SAVEDATE 1240
17.16.5.54 SECTION 1241
17.16.5.55 SECTIONPAGES 1241
17.16.5.56 SEQ 1242
17.16.5.57 SET 1243
17.16.5.58 SKIPIF 1244
17.16.5.59 STYLEREF 1245
17.16.5.60 SUBJECT 1246
17.16.5.61 SYMBOL 1247
17.16.5.62 TA 1248
17.16.5.63 TC 1249
17.16.5.64 TEMPLATE 1249
17.16.5.65 TIME 1250
17.16.5.66 TITLE 1250
17.16.5.67 TOA 1251
17.16.5.68 TOC 1252
17.16.5.69 USERADDRESS 1254
17.16.5.70 USERINITIALS 1255
17.16.5.71 USERNAME 1255
17.16.5.72 XE 1256
17.16.6 calcOnExit (Recalculate Fields When Current Field Is Modified) 1257
17.16.7 checkBox (Checkbox Form Field Properties) 1258
17.16.8 checked (Checkbox Form Field State) 1259
17.16.9 ddList (Drop-Down List Form Field Properties) 1259

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.10 default (Default Text Box Form Field String) 1260
17.16.11 default (Default Drop-Down List Item Index) 1261
17.16.12 default (Default Checkbox Form Field State) 1262
17.16.13 delInstrText (Deleted Field Code) 1263
17.16.14 enabled (Form Field Enabled) 1264
17.16.15 entryMacro (Script Function to Execute on Form Field Entry) 1265
17.16.16 exitMacro (Script Function to Execute on Form Field Exit) 1266
17.16.17 ffData (Form Field Properties) 1267
17.16.18 fldChar (Complex Field Character) 1268
17.16.19 fldSimple (Simple Field) 1272
17.16.20 format (Text Box Form Field Formatting) 1274
17.16.21 helpText (Associated Help Text) 1275
17.16.22 hyperlink (Hyperlink) 1277
17.16.23 instrText (Field Code) 1280
17.16.24 label (Form Field Label) 1281
17.16.25 listEntry (Drop-Down List Entry) 1283
17.16.26 maxLength (Text Box Form Field Maximum Length) 1284
17.16.27 name (Form Field Name) 1285
17.16.28 result (Drop-Down List Selection) 1286
17.16.29 size (Checkbox Form Field Size) 1288
17.16.30 sizeAuto (Automatically Size Form Field) 1288
17.16.31 statusText (Associated Status Text) 1289
17.16.32 tabIndex (Form Field Navigation Order Index) 1291
17.16.33 textInput (Text Box Form Field Properties) 1292
17.16.34 type (Text Box Form Field Type) 1292

17.17 Miscellaneous Topics 1293

17.17.1 Subdocuments 1293
17.17.1.1 subDoc (Anchor for Subdocument Location) 1295
17.17.2 Alternative Format Import 1298
17.17.2.1 altChunk (Anchor for Imported External Content) 1299
17.17.2.2 altChunkPr (External Content Import Properties) 1300
17.17.2.3 matchSrc (Keep Source Formatting on Import) 1301
17.17.3 Roundtripping Alternate Content 1302
17.17.4 Boolean Property (CT_OnOff) 1302

17.18 Simple Types 1302

17.18.1 ST_AnnotationVMerge (Table Cell Vertical Merge Revision Type) 1302
17.18.2 ST_Border (Border Styles) 1303
17.18.3 ST_BrClear (Line Break Text Wrapping Restart Location) 1356
17.18.4 ST_BrType (Break Types) 1358
17.18.5 ST_CaptionPos (Automatic Caption Positioning Values) 1359
17.18.6 ST_ChapterSep (Chapter Separator Types) 1361
17.18.7 ST_CharacterSpacing (Character-Level Whitespace Compression Settings) 1362
17.18.8 ST_CombineBrackets (Two Lines in One Enclosing Character Type) 1362
17.18.9 ST_DateTime (Standard Date and Time Storage Format) 1363
17.18.10 ST_DecimalNumber (Decimal Number Value) 1364
17.18.11 ST_DecimalNumberOrPercent (Percentage Measurement) 1364
17.18.12 ST_Direction (Bidirectional Direction Types) 1365

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.18.13 ST_DisplacedByCustomXml (Location of Custom XML Markup Displacing an Annotation) 1365
17.18.14 ST_DocGrid (Document Grid Types) 1366
17.18.15 ST_DocPartBehavior (Insertion Behavior Types) 1368
17.18.16 ST_DocPartGallery (Entry Gallery Types) 1369
17.18.17 ST_DocPartType (Entry Types) 1372
17.18.18 ST_DocProtect (Document Protection Types) 1374
17.18.19 ST_DocType (Document Classification Values) 1375
17.18.20 ST_DropCap (Text Frame Drop Cap Location) 1375
17.18.21 ST_EdGrp (Range Permision Editing Group) 1376
17.18.22 ST_EdnPos (Endnote Positioning Location) 1377
17.18.23 ST_EighthPointMeasure (Measurement in Eighths of a Point) 1378
17.18.24 ST_Em (Emphasis Mark Type) 1378
17.18.25 ST_FFHelpTextVal (Help Text Value) 1380
17.18.26 ST_FFName (Form Field Name Value) 1381
17.18.27 ST_FFStatusTextVal (Status Text Value) 1381
17.18.28 ST_FFTextType (Text Box Form Field Type Values) 1382
17.18.29 ST_FldCharType (Complex Field Character Type) 1383
17.18.30 ST_FontFamily (Font Family Value) 1384
17.18.31 ST_FrameLayout (Frameset Layout Order) 1385
17.18.32 ST_FrameScrollbar (Frame Scrollbar Visibility) 1386
17.18.33 ST_FtnEdn (Footnote or Endnote Type) 1387
17.18.34 ST_FtnPos (Footnote Positioning Location) 1389
17.18.35 ST_HAnchor (Horizontal Anchor Location) 1390
17.18.36 ST_HdrFtr (Header or Footer Type) 1391
17.18.37 ST_HeightRule (Height Rule) 1392
17.18.38 ST_HexColor (Color Value) 1393
17.18.39 ST_HexColorAuto ('Automatic' Color Value) 1393
17.18.40 ST_HighlightColor (Text Highlight Colors) 1394
17.18.41 ST_Hint (Font Type Hint) 1396
17.18.42 ST_HpsMeasure (Measurement in Half-Points) 1397
17.18.43 ST_InfoTextType (Help or Status Text Type) 1397
17.18.44 ST_Jc (Horizontal Alignment Type) 1398
17.18.45 ST_JcTable (Table Alignment Type) 1401
17.18.46 ST_LevelSuffix (Content Between Numbering Symbol and Paragraph Text) 1402
17.18.47 ST_LineNumberRestart (Line Numbering Restart Position) 1403
17.18.48 ST_LineSpacingRule (Line Spacing Rule) 1404
17.18.49 ST_Lock (Locking Types) 1404
17.18.50 ST_LongHexNumber (Eight Digit Hexadecimal Value) 1406
17.18.51 ST_MacroName (Script Subroutine Name Value) 1406
17.18.52 ST_MailMergeDataType (Mail Merge Data Source Type Values) 1407
17.18.53 ST_MailMergeDest (Merged Document Destination Types) 1408
17.18.54 ST_MailMergeDocType (Source Document Types) 1409
17.18.55 ST_MailMergeOdsoFMDFieldType (Merge Field Mapping Types) 1409
17.18.56 ST_MailMergeSourceType (Mail Merge ODSO Data Source Types) 1410
17.18.57 ST_Merge (Merged Cell Type) 1411
17.18.58 ST_MultiLevelType (Numbering Definition Type) 1414
17.18.59 ST_NumberFormat (Numbering Format) 1414
17.18.60 ST_ObjectDrawAspect (Embedded Object Representations) 1458

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.18.61 ST_ObjectUpdateMode (Embedded Object Update Modes) 1458
17.18.62 ST_PageBorderDisplay (Page Border Display Options) 1458
17.18.63 ST_PageBorderOffset (Page Border Positioning Base) 1459
17.18.64 ST_PageBorderZOrder (Page Border Z-Order) 1460
17.18.65 ST_PageOrientation (Page Orientation) 1461
17.18.66 ST_Pitch (Font Pitch Value) 1461
17.18.67 ST_PixelsMeasure (Measurement in Pixels) 1462
17.18.68 ST_PointMeasure (Measurement in Points) 1462
17.18.69 ST_Proof (Proofing State Values) 1462
17.18.70 ST_ProofErr (Proofing Error Type) 1463
17.18.71 ST_PTabAlignment (Absolute Position Tab Alignment) 1464
17.18.72 ST_PTabLeader (Absolute Position Tab Leader Character) 1465
17.18.73 ST_PTabRelativeTo (Absolute Position Tab Positioning Base) 1467
17.18.74 ST_RestartNumber (Footnote/Endnote Numbering Restart Locations) 1467
17.18.75 ST_RubyAlign (Phonetic Guide Text Alignment) 1468
17.18.76 ST_SdtDateMappingType (Date Storage Format Types) 1470
17.18.77 ST_SectionMark (Section Type) 1471
17.18.78 ST_Shd (Shading Patterns) 1474
17.18.79 ST_ShortHexNumber (Four Digit Hexadecimal Value) 1488
17.18.80 ST_SignedHpsMeasure (Signed Measurement in Half-Points) 1489
17.18.81 ST_SignedTwipsMeasure (Signed Measurement in Twentieths of a Point) 1489
17.18.82 ST_StyleSort (Style Sort Settings) 1490
17.18.83 ST_StyleType (Style Types) 1491
17.18.84 ST_TabJc (Custom Tab Stop Type) 1492
17.18.85 ST_TabTlc (Custom Tab Stop Leader Character) 1493
17.18.86 ST_TargetScreenSz (Target Screen Sizes for Generated Web Pages) 1495
17.18.87 ST_TblLayoutType (Table Layout Type) 1496
17.18.88 ST_TblOverlap (Table Overlap Setting) 1499
17.18.89 ST_TblStyleOverrideType (Conditional Table Style Formatting Types) 1501
17.18.90 ST_TblWidth (Table Width Units) 1502
17.18.91 ST_TextAlignment (Vertical Text Alignment Types) 1503
17.18.92 ST_TextboxTightWrap (Lines To Tight Wrap Within Text Box) 1504
17.18.93 ST_TextDirection (Text Flow Direction) 1505
17.18.94 ST_TextEffect (Animated Text Effects) 1507
17.18.95 ST_TextScale (Text Expansion/Compression Percentage) 1508
17.18.96 ST_Theme (Theme Font) 1508
17.18.97 ST_ThemeColor (Theme Color) 1509
17.18.98 ST_UcharHexNumber (Two Digit Hexadecimal Value) 1511
17.18.99 ST_Underline (Underline Patterns) 1511
17.18.100 ST_VAnchor (Vertical Anchor Location) 1515
17.18.101 ST_VerticalJc (Vertical Alignment Type) 1516
17.18.102 ST_View (Document View Values) 1517
17.18.103 ST_WmlColorSchemeIndex (Theme Color Reference) 1518
17.18.104 ST_Wrap (Text Wrapping around Text Frame Type) 1519
17.18.105 ST_Zoom (Magnification Preset Values) 1521
17.18.106 ST_TextScalePercent (Text Expansion/Compression Percentage) 1521
17.18.107 ST_MeasurementOrPercent (Measurement or Percentage Value) 1522

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

End of informative text.

17.2 Main Document Story

As defined in §4, a WordprocessingML document contains the markup for a rendition of an Office Open XML document of category Wordprocessing. Syntactically, the document consists of a compilation of two kinds of information, which are combined to create this rendition:

- Properties [*Example*: styles, numbering definitions, etc. *end example*]
- Stories [*Example*: main document, comments, headers, etc. *end example*]

In WordprocessingML, *stories* are unique containers for one or more paragraphs, as defined by the parent elements of the p element (§17.3.1.22). Stories contain the document's content. The properties of the document are applied to the contents of each story to create the rendition. Most of the content in a WordprocessingML document is located in the main document story, which is stored inside the body element within the Main Document part (§11.3.10).

[*Example*: Consider a document with a single paragraph in the main document story. This document would require the following WordprocessingML in its main document part:

```
<w:document>
  <w:body>
    <w:p/>
  </w:body>
</w:document>
```

The fact that the paragraph is inside the body element makes it part of the main document story. *end example*]

17.2.1 background (Document Background)

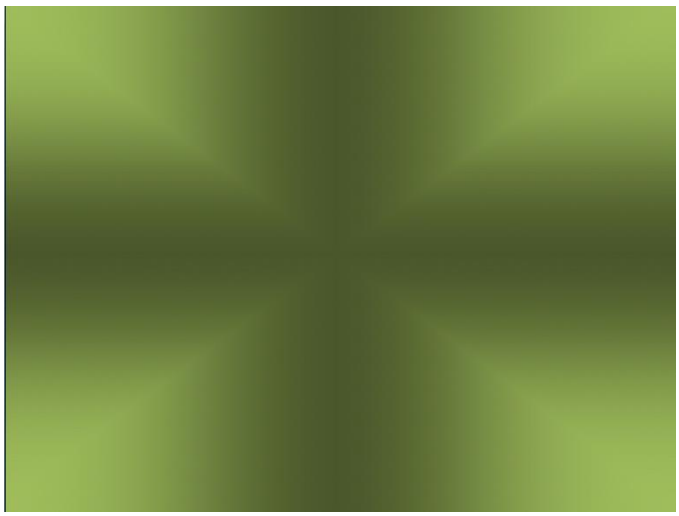
This element specifies the background for every page of the document containing the background element. A document's *background* is the image or fill for the entire page surface, behind all other document content.

The drawing §17.3.3.9 child elements of the background element allows any DrawingML effect to be applied to the document's background.

For solid color fill backgrounds, however, the attributes on this element allow the use of any RGB or theme color value (the latter a reference to the document's themes part).

[*Example*: Consider a document which utilizes a gradient fill background moving between black and the accent3 theme color, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



This background would require the following WordprocessingML markup:

```
<w:background w:themeColor="accent3">

  <w:drawing>
    <wp:inline … >
      <a:graphic>
        <a:graphicData … >
          …
        </a:graphicData>
      </a:graphic>
    </wp:inline>
  </w:drawing></w:background>
```

The resulting background consists of a single color fill of the `accent3` theme color from the themeColor attribute, layered under a `gradientCenter` fill. *end example*]

Attributes	Description
color (Background Color)	Specifies the color for the background of the document. This value can be defined as either: • A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*]. RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the background color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If the background specifies the use of a theme color via the themeColor attribute, this value is ignored. [*Note*: Applications are discouraged from specifying both the color and themeColor attributes on the same parent element. *end note*] If neither the color nor themeColor attributes are present, the parent page shall be treated as though it has no background defined. [*Example*: Consider a background color with value 2C34FF, as follows: `<w:background … w:color="2C34FF"/>` The background color is therefore the color with RGB value 44,52,255 (the decimal decoding of the hex value above). *end example*] The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).

themeColor (Background Theme Color)	Specifies the base theme color used to generate the background color. The background color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to be set centrally in the document.
	
	If the color attribute is specified, its value shall be ignored in favor of the color resulting from the use of this attribute with any appropriate themeTint and themeShade attribute value calculations applied.
	
	To determine the color to display, the following actions are performed:
	• Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read.
	• Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color.
	• The specified color is modified based on the presence of the themeTint or themeShade attribute.
	
	[*Example*: Consider a background configured to use the accent5 theme color, resulting in the following WordprocessingML markup:
	
	`<w:background w:themeColor="accent5" />`
	
	If the Settings part contained the following markup:
	
	`<w:clrSchemeMapping … w:accent5="accent5"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	and the Theme part contained the following XML markup:
	
	`<a:accent5>`
	`  <a:srgbClr val="BCBCBC"/>`
	`</a:accent5>`
	
	the resulting background color would be BCBCBC. *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Background Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this background. If the themeColor attribute is not specified, this attribute shall not be specified. If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this background. If the themeTint is supplied, the value of this attribute shall be ignored. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current background. [*Example*: Consider a shade of 60% applied to a background in a document. This shade is calculated as follows: $$S_{xml} = 0.6 * 255 = 153 = 99(hex)$$ The resulting themeShade value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * Shade_{percentage}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is C0504D. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get 943634. This transformed value can be seen in the resulting background's color attribute: `<w:background w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Background Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this background. If the themeColor attribute is not specified, this attribute shall not be specified. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to the document's background. The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current background. [*Example*: Consider a tint of 60% applied to a background in a document. This tint is calculated as follows: $$T_{xml} = 0.4 * 255 \\ = 102 \\ = 66(hex)$$ The resulting themeTint value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L'} = \mathrm{L} * \mathrm{Tint_{pct}} + \left(1 - \mathrm{Tint_{pct}}\right)$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	whose RGB value (in RRGGBB hex format) is 4F81BD. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6)$$ $$= 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get 95B3D7. This transformed value can be seen in the resulting background's color attribute: `<w:background w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Background) is located in §A.1. *end note*]

17.2.2 body (Document Body)

This element specifies the contents of the body of the document - the main document editing surface.

The document body contains what is referred to as *block-level markup* - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

[*Example*: Consider a document with a single paragraph in the main document story. This document would require the following WordprocessingML in its main document part:

```
<w:document>
  <w:body>
    <w:p/>
  </w:body>
</w:document>
```

The fact that the paragraph is inside the body element makes it part of the main document story. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Body) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.2.3 document (Document)

This element specifies the contents of a main document part in a WordprocessingML document.

[*Example*: Consider the basic structure of the main document part in a basic WordprocessingML document, as follows:

```
<w:document>
  <w:body>
    <w:p/>
  </w:body>
</w:document>
```

All of the contents of the main document part are contained beneath the document element. *end example*]

Attributes	Description

conformance (Document Conformance Class)	Specifies the conformance class (§2.1) to which the WordprocessingML document conforms.
	
	If this attribute is omitted, its default value is `transitional`.
	
	[*Example*: Consider the following WordprocessingML Main Document part markup:
	
	`<w:document w:conformance="strict">`
	`…`
	`</w:document>`
	
	This document has a conformance attribute value of `strict`, therefore it conforms to the WML Strict conformance class. *end example*]
	
	The possible values for this attribute are defined by the ST_ConformanceClass simple type (§22.9.2.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Document) is located in §A.1. *end note*]

17.3 Paragraphs and Rich Formatting

The basis of a WordprocessingML document is its actual text contents. Those text contents can be stored in many contexts (tables, text boxes, etc.), but the most basic form of text contents in WordprocessingML is the paragraph, specified using the p element (§17.3.1.22).

Within the paragraph, all rich formatting at the paragraph level is stored within the pPr element (§17.3.1.25; §17.3.1.26). [*Note*: Some examples of paragraph properties are alignment, border, hyphenation override, indentation, line spacing, shading, text direction, and widow/orphan control. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Within the paragraph, text is grouped into one or more runs, represented by the r element (§17.3.2.25), which define a region of text with a common set of properties.

Just as a paragraph can have rich formatting, so too can a run. All of the elements inside an r element have their properties controlled by a corresponding optional rPr run properties element (§17.7.9.1; §17.3.2.27). [*Note*: Some examples of run properties are bold, underlined, or visible. *end note*]

Within runs, run content is the set of possible objects and characters which can be displayed in the document.

17.3.1 Paragraphs

The most basic unit of block-level content within a WordprocessingML document, *paragraphs* are stored using the p element (§17.3.1.22). A paragraph defines a distinct division of content with a WordprocessingML document which begins on a new line.

[*Example*: Consider the paragraph fragment "*The quick brown fox jumped …* " which is centered on a paragraph. The justification property is a paragraph level property, and therefore is expressed on the paragraph properties as follows:

```
<w:p>
  <w:pPr>
    <w:jc w:val="center"/>
    <w:rPr>
      <w:i/>
    </w:rPr>
  </w:pPr>
  <w:r>
    <w:rPr>
      <w:i/>
    </w:rPr>
    <w:t xml:space="preserve">The quick brown fox jumped … </w:t>
  </w:r>
</w:p>
```

Notice that each run specifies the character formatting information for its contents, and the paragraph specifies the paragraph level formatting (the center-justification). It is also notable that since leading and trailing whitespace is not normally significant in XML; some runs require an attribute specifying that their whitespace is significant via the xml:space element. *end example*]

A paragraph's properties are specified via the pPr element (§17.3.1.25; §17.3.1.26). [*Note*: Some examples of paragraph properties are alignment, border, hyphenation override, indentation, line spacing, shading, text direction, and widow/orphan control. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.1.1 adjustRightInd (Automatically Adjust Right Indent When Using Document Grid)

This element specifies whether the right indent shall be automatically adjusted for the given paragraph when a document grid has been defined for the current section using the docGrid element (§17.6.5), modifying of the current right indent used on this paragraph.

[*Note*: This setting is used in order to ensure that the line breaking for that paragraph is not determined by the width of the final character on the line. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be `true`.

[*Example*: Consider a paragraph in which the right indent on the current paragraph should not be automatically determined based on the character pitch set in the document grid. This setting would be specified using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    …
    <w:adjustRightInd w:val="false" />
  </w:pPr>
  …
</w:p>
```

By explicitly setting the val to `false`, this paragraph uses its specified right indent settings regardless of the presence of the document grid for the parent section. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.2 autoSpaceDE (Automatically Adjust Spacing of Latin and East Asian Text)

This element specifies whether inter-character spacing shall automatically be adjusted between regions of Latin text and regions of East Asian text in the current paragraph. These regions shall be determined by the Unicode character values of the text content within the paragraph.

[*Note*: This property is used to ensure that the spacing between regions of Latin text and adjoining East Asian text is sufficient on each side such that the Latin text can be easily read within the East Asian text. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be `true`.

[*Example*: Consider a paragraph in which the spacing should not be automatically adjusted based on the presence of Latin and East Asian text. This setting would be specified using the following WordprocessingML:

```
<w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <w:pPr>
      …
      <w:autoSpaceDE w:val="false" />
    </w:pPr>
    …
  </w:p>
```

By explicitly setting the val to `false`, this paragraph must not automatically adjust the spacing of adjoining Latin and East Asian text. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.3 autoSpaceDN (Automatically Adjust Spacing of East Asian Text and Numbers)

This element specifies whether inter-character spacing shall automatically be adjusted between regions of numbers and regions of East Asian text in the current paragraph. These regions shall be determined by the Unicode character values of the text content within the paragraph.

[*Note*: This property is used to ensure that the spacing between regions of numbers and adjoining East Asian text is sufficient on each side such that the numbers can be easily read within the East Asian text. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be `true`.

[*Example*: Consider a paragraph in which the spacing should not be automatically adjusted based on the presence of numbers and East Asian text. This setting would be specified using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    …
    <w:autoSpaceDN w:val="false" />
  </w:pPr>
  …
</w:p>
```

By explicitly setting the val to `false`, this paragraph automatically adjusts the spacing of adjoining numbers and East Asian text. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.4 bar (Paragraph Border Between Facing Pages)

This element specifies the border which can be displayed on the inside edge of the paragraph when the parent's section settings specify that the section shall be printed using mirrored margins using the mirrorMargins

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

element (§17.15.1.57). [*Note*: This information is present in the WordprocessingML for the purposes of legacy document format compatibility, and it can be removed and/or ignored as required. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no bar border shall be applied to the current paragraph.

[*Example*: Consider the following paragraph's WordprocessingML definition for its paragraph borders:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
        w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
        w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
        w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
        w:themeColor="accent3" w:themeTint="99" />
      <w:bar w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
        w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>Sample paragraph.</w:t>
  </w:r>
</w:p>
```

This paragraph has a single line bar border as defined by the bar element. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.5 between (Paragraph Border Between Identical Paragraphs)

This element specifies the border which shall be displayed between each paragraph in a set of paragraphs which have the same set of paragraph border settings.

To determine if any two adjoining paragraphs should have a between border or an individual top and bottom border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is identical for all possible paragraphs borders, then the between border is displayed. Otherwise, each paragraph shall use its bottom and top border, respectively. If this border specifies a space attribute, that value is ignored - this border is always located at the bottom of each paragraph with an identical following paragraph, taking into account any space after the line pitch.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied between identical paragraphs.

[*Example*: Consider the following two paragraphs' WordprocessingML definition:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>First paragraph.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="0" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>Second paragraph.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Since the bottom paragraph border is different between the two paragraphs (the bottom space value goes from 1 to 0), these paragraphs do not use the between border, and instead paragraph one uses its bottom border, and paragraph two uses its top border. If those values were identical, then paragraph one would have a between border below it, and paragraph two would have no top border. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.6 bidi (Right to Left Paragraph Layout)

This element specifies that this paragraph shall be displayed from right to left. This property only affects the following set of paragraph-level properties:

- ind (§17.3.1.12)
- jc (§17.3.1.13)
- tab (§17.3.1.37)
- textDirection (§17.3.1.41)

This setting alone does not affect the ordering of text within the paragraph – see the rtl element (§17.3.2.30) for a detailed description.

This element specifies that the base direction of the text within the paragraph is right-to-left (see HL1 in the Unicode Standard Annex #9). See also Part 1, §I.2.

[*Example*: Consider a paragraph with the bidi property set as follows:

```
<w:p>
    <w:pPr>
      <w:bidi/>
    </w:pPr>
    …
</w:p>
```

This paragraph direction is now right to left, which means that all paragraph properties are displayed right to left (e.g. the paragraph marker glyph (if any) is displayed on the right, and indentation for the first line of the paragraph occurs on the right side of the page). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.7 bottom (Paragraph Border Below Identical Paragraphs)

This element specifies the border which shall be displayed below a set of paragraphs which have the same paragraph border settings.

To determine if any two adjoining paragraphs shall have an individual top and bottom border or a between border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is different, then the first paragraph shall use its bottom border and the following paragraph shall use its top border. Otherwise, the between border is used. If this border specifies a space attribute, that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

value determines the space after the bottom of the text (ignoring any space below) which should be left before this border is drawn, specified in points.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied below identical paragraphs.

[*Example*: Consider the following two paragraphs' WordprocessingML definition:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>First paragraph.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="0" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <w:t>Second paragraph.</w:t>
    </w:r>
  </w:p>
```

Since the paragraph border is different between the two paragraphs (the bottom space value goes from 1 to 0), paragraph one uses its bottom border, which is located one point below the text in that paragraph. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.8 cnfStyle (Paragraph Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this paragraph, if this paragraph is contained within a table cell. [*Note*: This property is an optimization which can be used by consumers to determine if a given property on a paragraph is the result of the table style properties vs. direct formatting on the paragraph itself. *end note*]

If this property is specified on a paragraph which is not contained within a table cell, then its contents shall be ignored when reading the contents of the document.

[*Example*: Consider a paragraph in the top right corner of a table with a table style applied and where the table is formatted as left to right. This paragraph would need to specify the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:cnfStyle w:firstRow="true" w:lastColumn="true"
w:firstRowLastColumn="true" />
    …
  </w:pPr>
  …
</w:p>
```

This paragraph specifies that it has the conditional properties from the table style for the first column, first row, and the top right corner of the parent table by setting the appropriate attributes. *end example*]

Attributes	Description

evenHBand (Even Numbered Horizontal Band)	Specifies that the object has inherited the conditional properties applied to the even numbered horizontal bands of the parent object.
	
	[*Example*: Consider a paragraph in the second row of a table with a table style applied, and where the band width is one row. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:evenHBand="true" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the even numbered horizontal bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
evenVBand (Even Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the even numbered vertical bands of the parent object. [*Example*: Consider a paragraph in the second column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:evenVBand="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the even numbered vertical bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstColumn (First Column)	Specifies that the object has inherited the conditional properties applied to the first column of the parent object. [*Example*: Consider a paragraph in the first column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:firstColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

firstRow (First Row)	Specifies that the object has inherited the conditional properties applied to the first row of the parent object.
	
	[*Example*: Consider a paragraph in the top row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the first row of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

firstRowFirstColumn (First Row and First Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and first column of the parent object.
	
	[*Example*: Consider a paragraph in the first row and first column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" w:firstColumn="true" w:firstRowFirstColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and first column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

firstRowLastColumn (First Row and Last Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and last column of the parent object.
	
	[*Example*: Consider a paragraph in the first row and last column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" w:lastColumn="true"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	w:firstRowLastColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and last column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastColumn (Last Column)	Specifies that the object has inherited the conditional properties applied to the last column of the parent object. [*Example*: Consider a paragraph in the last column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the last column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRow (Last Row)	Specifies that the object has inherited the conditional properties applied to the last row of the parent object. [*Example*: Consider a paragraph in the bottom row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastRow="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the last row of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

lastRowFirstColumn (Last Row and First Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the last row and first column of the parent object.
	
	[*Example*: Consider a paragraph in the last row and first column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastRow="true" w:firstColumn="true" w:lastRowFirstColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and first column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

lastRowLastColumn (Last Row and Last Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the last row and last column of the parent object.
	
	[*Example*: Consider a paragraph in the last row and last column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastRow="true" w:lastColumn="true" w:lastRowLastColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and last column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

oddHBand (Odd Numbered Horizontal Band)	Specifies that the object has inherited the conditional properties applied to the odd numbered horizontal bands of the parent object. [*Example*: Consider a paragraph in the third row of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML: `<w:p>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:cnfStyle w:oddHBand="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the odd numbered horizontal bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
oddVBand (Odd Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the odd numbered vertical bands of the parent object. [*Example*: Consider a paragraph in the third column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:oddVBand="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the odd numbered vertical bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cnf) is located in §A.1. *end note*]

17.3.1.9 contextualSpacing (Ignore Spacing Above and Below When Using Identical Styles)

This element specifies that any space specified before or after this paragraph, specified using the spacing element (§17.3.1.33), should not be applied when the preceding and following paragraphs are of the same paragraph style, affecting the top and bottom spacing respectively. [*Example*: This value is typically used for paragraphs in lists, in which any space between subsequent list items, even if inherited from another style, is not desirable. *end example*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then spacing is not ignored. If it is present, then the spacing above or below on this paragraph is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

subtracted from the spacing which would have been present if contextual spacing was off, never going below zero.

[*Example*: Consider two paragraphs defined as follows:

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="TestParagraphStyle" />
    <w:spacing w:after="200"/>
    <w:contextualSpacing/>
  </w:pPr>
  …
</w:p>
<w:p>
  <w:pPr>
    <w:pStyle w:val="TestParagraphStyle" />
    <w:spacing w:before="240"/>
  </w:pPr>
  …
</w:p>
```

The first paragraph specifies a spacing after of 10 points, and the second paragraph specifies a spacing before of 12 points, therefore according to the rules on the spacing element, the net paragraph spacing should be 12 points. However, since the first paragraph specifies that its spacing should be omitted between paragraphs of the same style, and the two paragraphs use the same `TestParagraphStyle`, that value is subtracted from the total, therefore the paragraphs are spaced by 2 points. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.10 divId (Associated HTML div ID)

This element specifies that this paragraph should be located within the specified HTML `div` tag when this document is saved in HTML format. This ID is then used to look up the associated div stored in the divs (§17.15.2.8) element. [*Note*: This element is used to preserve the fidelity of existing HTML documents when saved in the WordprocessingML format. *end note*].

If the paragraph does not specify this element, then any `div` referenced by the previous paragraph is closed, and this paragraph shall not belong to any `div` when saved as HTML. If this specified id does not exist in the collection of `div`s the current document, then any `div` referenced by the previous paragraph is closed, and this paragraph shall not belong to any `div` when saved as HTML.

[*Example*: Consider the following WordprocessingML paragraph fragment:

```
<w:p>
  <w:pPr>
    <w:divId w:val="1512645511" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </w:pPr>
  </w:p>
```

This paragraph specifies that it belongs to the HTML `div` with id `1512645511`, stored in the divs element. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.3.1.11 framePr (Text Frame Properties)

This element specifies information about the current paragraph with regard to *text frames*. *Text frames* are paragraphs of text in a document which are positioned in a separate region or frame in the document, and can be positioned with a specific size and position relative to non-frame paragraphs in the current document.

The first piece of information specified by the framePr element is that the current paragraph is actually part of a text frame in the document. This information is specified simply by the presence of the framePr element in paragraph's properties. If the framePr element is omitted, the paragraph shall not be part of any text frame in the document.

The second piece of information concerns the set of paragraphs which are part of the current text frame in the document. This is determined based on the attributes on the framePr element. If the set of attribute values specified on two adjacent paragraphs is identical, then those two paragraphs shall be considered to be part of the same text frame and rendered within the same frame in the document.

[*Example*: Consider a document in which the following two paragraphs are located adjacent to one another:

```
<w:p>
  <w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:framePr w:w="2191" w:h="811" w:hRule="exact" w:hSpace="180"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1921"/>
  </w:pPr>
  <w:r>
    <w:t>Paragraph One</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:framePr w:w="2191" w:h="810" w:hRule="exact" w:hSpace="180"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1921"/>
  </w:pPr>
  <w:r>
    <w:t>Paragraph Two.</w:t>
  </w:r>
</w:p>
```

These two paragraphs, although each is a part of a text frame due to the presence of the framePr element, are different text frames because of the differing h value - 810 vs. 811. *end example*]

The positioning of the frame relative to the properties stored on its attribute values shall be calculated relative to the next paragraphs in the document which is itself not part of a text frame.

[*Example*: Consider a document in which the following three paragraphs are located adjacent to one another:

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2191" w:h="811" w:hRule="exact" w:hSpace="180"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1921"/>
  </w:pPr>
  <w:r>
    <w:t>Paragraph One</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:framePr w:w="2191" w:h="811" w:hRule="exact" w:hSpace="180"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1921"/>
  </w:pPr>
  <w:r>
    <w:t>Paragraph Two.</w:t>
  </w:r>
</w:p>
<w:p/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The first two paragraphs form a single text frame, which is anchored using its attribute values relative to the first non-frame paragraph following it (the third paragraph in the example). *end example*]

Attributes	Description

anchorLock (Lock Frame Anchor to Paragraph)	Specifies that the frame shall always remain in the same logical position relative to the non-frame paragraphs which precede and follow it in this document.
	
	This means that consumers which modify this document shall ensure that this text frame remains directly above the non-frame paragraph which it is currently above, by adjusting the frame's positioning properties as needed as the paragraph is moved throughout the document rather than moving the frame's logical location within the paragraphs in the document, if that would be more appropriate.
	
	If this attribute is omitted, then this frame shall not have a locked anchor position.
	
	[*Example*: Consider the following WordprocessingML paragraph contained in a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="exact" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="73" w:anchorLock="1"/>`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text frame has a locked anchor using the anchorLock attribute. If the text frame is moved down in the document, the text frame properties must be adjusted to be relative to the parent paragraph's same logical position - the paragraph cannot be relocated in the document, which results in changes to the frame's properties as follows:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="exact" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="-5247" w:anchorLock="1"/>`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	The non-frame paragraph was relocated 5320 twentieths of a point below its original location in the document, and the frame's vertical positioning properties were adjusted to ensure its logical location within the paragraph ordering was constant while its visual

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	location was changed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

dropCap (Drop Cap Frame)	Specifies that the current frame contains a drop cap to be located at the beginning of the next non-frame paragraph in the document. Its contents shall be used to specify how that drop cap should be positioned relative to that paragraph.
	
	If this attribute is omitted, then this frame shall not be considered a drop cap frame.
	
	[*Note*: Although a drop cap is simply a text frame, this element is used to determine how the cap should be positioned relative to the following non-frame paragraph in relative terms (see possible values), rather than relying on absolute sizing. *end note*]
	
	[*Example*: Consider the following paragraph containing a text frame which should be positioned as a drop cap:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:dropCap="margin" w:lines="3" w:hSpace="432" w:wrap="around" w:vAnchor="text" w:hAnchor="page" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>A</w:t>`
	`  </w:r>`
	`</w:p>`
	
	The dropCap attribute specifies a value of `margin`, so this drop cap is placed outside of the text margin before the start of the current text. *end example*]
	
	The possible values for this attribute are defined by the ST_DropCap simple type (§17.18.20).

h (Frame Height)	Specifies the frame's height. This height is expressed in twentieths of a point. If this attribute is omitted, then its value shall be assumed to be `0`. The meaning of the value of the h attribute is defined based on the value of the hRule attribute for this text frame as follows: • If the value of hRule is `auto`, then the frame's height should be automatically determined based on the height of its contents. This value is ignored. • If the value of hRule is `atLeast`, then the frame's height should be at least the value of this attribute. • If the value of hRule is `exact`, then the frame's height should be exactly the value of this attribute.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider the following paragraph containing a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast"`
	`w:hSpace="187" w:wrap="around" w:vAnchor="text"`
	`w:hAnchor="page" w:x="1643" w:y="73" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	The h attribute specifies a value of 2189 twentieths of a point, so this text frame is a minimum of 2189 twentieths of a point high regardless of its contents, since its hRule value is set to `atLeast`. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

hAnchor (Frame Horizontal Positioning Base)	Specifies the base object from which the horizontal positioning in the x attribute should be calculated.
	
	A text frame can be horizontally positioned relative to:
	• The vertical edge of the page before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs)
	• The vertical edge of the text margin before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs)
	• The vertical edge of the text margin for the column in which the anchor paragraph is located
	
	If this attribute is omitted, then its value shall be assumed to be page.
	
	[*Example*: Consider a text frame which should be positioned one inch to the right of its column in a left-to-right document. This text frame would be specified using the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:framePr … w:x="1440" w:hAnchor="column" />`
	`</w:pPr>`
	
	These frame properties specify that they are relative to the anchor paragraph's `column`, and that relative to that column, the frame should be `1440` twentieths of a point in the direction of the flow of text (right, in this case). *end example*]
	
	The possible values for this attribute are defined by the ST_HAnchor simple type (§17.18.35).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

hRule (Frame Height Type)	Specifies the meaning of the height specified for this frame.
	
	The meaning of the value of the h attribute is defined based on the value of the hRule attribute for this text frame as follows:
	• If the value of hRule is `auto`, then the frame's height should be automatically determined based on the height of its contents. The h value is ignored.
	• If the value of hRule is `atLeast`, then the frame's height should be at least the value the h attribute.
	• If the value of hRule is `exact`, then the frame's height should be exactly the value of the h attribute.
	
	If this attribute is omitted, then its value shall be assumed to be auto.
	
	[*Example*: Consider the following paragraph containing a text frame:
	
	`<w:p>`
	`<w:pPr>`
	`<w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="73" />`
	`</w:pPr>`
	`<w:r>`
	`<w:t>Text Frame Content.</w:t>`
	`</w:r>`
	`</w:p>`
	
	The h attribute specifies a value of 2189 twentieths of a point, so this text frame is a minimum of 2189 twentieths of a point high regardless of its contents, since its hRule value is set to `atLeast`. *end example*]
	
	The possible values for this attribute are defined by the ST_HeightRule simple type (§17.18.37).

hSpace (Horizontal Frame Padding)	Specifies the minimum distance which shall be maintained between the current text frame and any non-frame text which has been allowed to flow around this object when the wrap attribute on this text frame is set to around. This distance is expressed in twentieths of a point. If the wrap value is not set to around, this value shall be ignored. If this attribute is omitted, its value shall be assumed to be `0`. [*Example*: Consider a text frame which should have a minimum of a one-half inch spacing from any non-frame text on its left and right sides. This constraint would be specified using the following WordprocessingML: `<w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:framePr … w:hSpace="720" w:wrap="around" /> </w:pPr> The wrap value of around allows text to wrap around this text frame, and the hSpace attribute specifies that the spacing between text and this frame must be a minimum of 720 twentieths of a point. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
lines (Drop Cap Vertical Height in Lines)	Specifies the number of lines in the non-frame paragraph to which this text frame is anchored which should be used to calculate the drop cap's height. If the current frame is not a drop cap (the parent framePr element does not have the dropCap attribute), this value is ignored. If the current text frame is a dropped cap and this attribute is present, then any other vertical positioning information shall be ignored. If this attribute is omitted, then its value shall be considered to be 1. [*Example*: Consider the following paragraph containing a text frame which should be positioned as a drop cap: <w:p> <w:pPr> <w:framePr w:dropCap="margin" w:lines="3" w:hSpace="432" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:y="400" w:yAlign="text" /> </w:pPr> <w:r> <w:t>O</w:t> </w:r> </w:p> Since this frame is being used as a dropped cap, the y and yAlign attributes are ignored and the height of the drop cap is the first three lines of the anchor paragraph. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
vAnchor (Frame Vertical Positioning Base)	Specifies the base object from which the horizontal positioning in the y attribute should be calculated. A text frame can be horizontally positioned relative to: • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the text margin before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	• The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections)
	
	If this attribute is omitted, then its value shall be assumed to be page.
	
	[*Example*: Consider a text frame which should be positioned two inches below the page top in a top-to-bottom document. This text frame would be specified using the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:framePr … w:y="2880" w:vAnchor="page" />`
	`</w:pPr>`
	
	These frame properties specify that they are relative to the anchor page, and that relative to that column, the frame should be `2880` twentieths of a point in the direction of the flow of text (down, in this case). *end example*]
	
	The possible values for this attribute are defined by the ST_VAnchor simple type (§17.18.100).

vSpace (Vertical Frame Padding)	Specifies the minimum distance which shall be maintained between the current text frame and any non-frame text which is above or below this text frame.
	
	This distance is expressed in twentieths of a point.
	
	If this attribute is omitted, its value shall be assumed to be `0`.
	
	[*Example*: Consider a text frame which should have a minimum of a one-half inch spacing from any non-frame text on its top and bottom sides. This constraint would be specified using the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:framePr … w:vSpace="720" />`
	`</w:pPr>`
	
	The vspace attribute specifies that the spacing between text and this frame must be a minimum of `720` twentieths of a point. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

w (Frame Width)	Specifies the exact value for this text frame's width. This value is specified in twentieths of a point. When this attribute is present, the text frame shall be rendered to the exact width specified. If this attribute is omitted, the text frame width shall be automatically

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	determined by the maximum line width of the content within the text frame.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast"`
	`w:hSpace="187" w:wrap="around" w:vAnchor="text"`
	`w:hAnchor="page" w:x="1643" w:y="73" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text frame specifies that its width must be exactly 2419 twips. If this attribute was removed, the text frame would be rendered at the width of the content `Text Frame Content`. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

wrap (Text Wrapping Around Frame)	Specifies the style of text wrapping which should be allowed around the contents of this text frame. This attribute determines if non-frame text shall be allowed to flow around the contents of this frame.
	
	If this attribute is omitted, its value shall be assumed to be `around`.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast"`
	`w:hSpace="187" w:wrap="around" w:vAnchor="text"`
	`w:hAnchor="page" w:x="1643" w:y="73" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text frame specifies that when the frame is rendered on the page, any non-text frame paragraphs which would normally flow onto the same lines must be allowed to do so. *end example*]
	
	The possible values for this attribute are defined by the ST_Wrap simple type (§17.18.104).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

x (Absolute Horizontal Position)	Specifies an absolute horizontal position for the text frame. This absolute position is specified relative to the horizontal anchor specified by the hAnchor attribute for this text frame.
	
	This value is expressed in twentieths of a point. If it is positive, then the text frame is positioned after the anchor object in the direction of horizontal text flow in this document. If it is negative, then the text frame is positioned before the anchor object in the direction of horizontal text flow in this document.
	
	If the xAlign attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="73" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text frame specifies that it should be located exactly 1643 twentieths of a point after the vertical edge of the page (from the hAnchor attribute). *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

xAlign (Relative Horizontal Position)	Specifies a relative horizontal position for the text frame. This relative position is specified relative to the horizontal anchor specified by the hAnchor attribute for this text frame.
	
	If omitted, this attribute is not specified and the value of the x attribute determines the absolute horizontal position of the text frame. If specified, the position for this attribute supersede any value which is specified in the x attribute, and that value is ignored.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:xAlign="left" w:y="73" />`
	`  </w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:r> <w:t>Text Frame Content.</w:t> </w:r> </w:p> This text frame specifies that it has a horizontal placement of exactly 1643 twentieths of a point relative to the page, but that exact placement is overridden by the presence of the xAlign attribute to place the frame on the left side of the page. *end example*] The possible values for this attribute are defined by the ST_XAlign simple type (§22.9.2.18).
y (Absolute Vertical Position)	Specifies an absolute vertical position for the text frame. This absolute position is specified relative to the vertical anchor specified by the vAnchor attribute for this text frame. This value is expressed in twentieths of a point. If it is positive, then the text frame is positioned after the anchor object in the direction of vertical text flow in this document. If it is negative, then the text frame is positioned before the anchor object in the direction of vertical text flow in this document. If the yAlign attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0. [*Example*: Consider the following WordprocessingML fragment specifying a text frame: <w:p> <w:pPr> <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187" w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="73" /> </w:pPr> <w:r> <w:t>Text Frame Content.</w:t> </w:r> </w:p> This text frame specifies that it should be located exactly 79 twentieths of a point below the top vertical edge of the anchor's paragraph's text (from the vAnchor attribute), assuming that the vertical text direction is top to bottom. *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).
yAlign (Relative Vertical Position)	Specifies a relative vertical position for the text frame. This relative position is specified relative to the vertical anchor specified by the vAnchor attribute for this text frame. If omitted, this attribute is not specified and the value of the y attribute determines the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	absolute horizontal position of the text frame. If specified, the position for this attribute supersedes any value which is specified in the y attribute, and that value is ignored, unless the vAnchor is set to text, in which case any relative positioning is not allowed, and is itself ignored.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a text frame:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187" w:wrap="around" w:vAnchor="margin" w:hAnchor="page" w:x="1643" w:y="73" w:yAlign="center" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Text Frame Content.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text frame specifies that it has a vertical placement of exactly 73 twentieths of a point relative to the top margin, but that exact placement is overridden by the presence of the yAlign attribute to place the frame in the center of the margin. *end example*]
	
	The possible values for this attribute are defined by the ST_YAlign simple type (§22.9.2.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FramePr) is located in §A.1. *end note*]

17.3.1.12 ind (Paragraph Indentation)

This element specifies the set of indentation properties applied to the current paragraph.

Indentation settings are overriden on an individual basis - if any single attribute on this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If any single attribute on this element is never specified in the style hierarchy, then no indentation of that indentation type is applied to the paragraph.

[*Example*: Consider a paragraph which should have a one inch indentation from the text margins on both the left and the right sides, except for the first line in each paragraph, which should only be indented one quarter of an inch from the text margin (on the side which begins the flow of text for this paragraph). This set of indentations is specified using the following WordprocessingML:

```
<w:pPr>
  <w:ind w:start="1440" w:end="1440" w:hanging="1080" />
</w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This set of indentation properties specifies that a 1440 twentieths of a point indentation should be provided on both the left and the right side of the text margins for this paragraph, and that a 1080 twentieths of a point hanging indent (towards the text margin) should be applied to the text in the first paragraph, giving it a net one-quarter inch indent from the text margin. *end example*]

Attributes	Description

end (End Indentation)	Specifies the indentation which shall be placed at the end of this paragraph – between the right text margin for this paragraph and the right edge of that paragraph's content in a left to right paragraph, and the left text margin and the left edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§17.3.1.18) is specified for this paragraph, then this indent is used for the outside page edge - the left page edge for odd numbered pages and the right page edge for even numbered pages.
	
	If this attribute is omitted, its value shall be assumed to be zero.
	
	All other values for this element are relative to the trailing text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. As well, if the endChars attribute is specified, then this value is ignored.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="720" w:end="-1440" />`
	`</w:pPr>`
	
	This set of paragraph indentations specifies that this paragraph's text should be indented 1440 twentieths of a point (one inch) into the right text margin in this document, assuming this is a left to right paragraph. *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

endChars (End Indentation in Character Units)	Specifies the indentation which shall be placed at the end of this paragraph – between the right text margin for this paragraph and the right edge of that paragraph's content in a left to right paragraph, and the left text margin and the left edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§17.3.1.18) is specified for this paragraph, then this indent is used for the outside page edge - the left page edge for odd numbered pages and the right page edge for even numbered pages. This value is specified in hundredths of a character unit. If this attribute is omitted, its value shall be assumed to be zero. All other values for this element are relative to the trailing text margin, negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. As well, if the end attribute is specified, then its value is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	ignored, and is superseded by this value.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:endChars="250" />`
	`</w:pPr>`
	
	This set of paragraph indentations specifies that this paragraph's text should be indented two and a half character units from the right text margin in this document, assuming this is a left to right paragraph. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

firstLine (Additional First Line Indentation)	Specifies the additional indentation which shall be applied to the first line of the parent paragraph. This additional indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph.
	
	The firstLine and hanging attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the firstLineChars attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be zero (if needed).
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="1440" w:end="720" w:firstLine="1440" />`
	`</w:pPr>`
	
	This set of indentations specifies that the first line should be indented 1440 twentieths of a point (one inch) from the indentation specified for all remaining paragraphs, which is the 1440 twentieths of a point, as specified by the start attribute. This gives the first line a two inch indentation from the text margin. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

firstLineChars (Additional First Line Indentation in Character Units)	Specifies the additional indentation which shall be applied to the first line of the parent paragraph. This additional indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph. It is specified in one hundredths of a character unit. The firstLineChars and hangingChars attributes are mutually exclusive, if both are specified, then the firstLineChars value is ignored. If the firstLine attribute is also specified, then this value supersedes its other value. If this attribute is omitted, then its value shall be assumed to be zero (if needed).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="1440" w:end="720" w:firstLineChars="140" />`
	`</w:pPr>`
	
	This set of indentations specifies that the first line should be indented 140 hundredths of a character units from the indentation specified for all remaining paragraphs, which is the 1440 twentieths of a point specified by the start attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

hanging (Indentation Removed from First Line)	Specifies the indentation which shall be removed from the first line of the parent paragraph, by moving the indentation on the first line back towards the beginning of the direction of text flow.
	
	This indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph.
	
	The firstLine and hanging attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the hangingChars attribute is also specified, then this value is ignored. If this attribute is omitted, its value shall be assumed to be zero (if needed).
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="1440" w:end="720" w:hanging="720" />`
	`</w:pPr>`
	
	This set of indentations specifies that the first line should be indented 720 twentieths of a point (one inch) towards the text margin from the indentation specified for all remaining paragraphs, which is the 1440 twentieths of a point specified by the start attribute. This gives the first line a one-half inch indentation from the text margin. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

hangingChars (Indentation Removed From First Line in Character Units)	Specifies the indentation which shall be removed from the first line of the parent paragraph, by moving the indentation on the first line back towards the beginning of the direction of text flow. This indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph. It is specified in one hundredths of a character unit.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	The firstLineChars and hangingChars attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the hanging attribute is also specified, then its value is superseded by this value. If this attribute is omitted, its value shall be assumed to be zero (if needed).
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="1440" w:end="720" w:hangingChars="100" />`
	`</w:pPr>`
	
	This set of indentations specifies that the first line should be indented one character unit towards the text margin from the indentation specified for all remaining paragraphs, which is the 1440 twentieths of a point specified by the start attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

start (Start Indentation)	Specifies the indentation which shall be placed at the start of this paragraph – between the left text margin for this paragraph and the left edge of that paragraph's content in a left to right paragraph, and the right text margin and the right edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§17.3.1.18) is specified for this paragraph, then this indent is used for the inside page edge - the right page edge for odd numbered pages and the left page edge for even numbered pages.
	
	If this attribute is omitted, its value shall be assumed to be zero.
	
	All other values for this element are relative to the leading text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. This value can be superseded for the first line only via use of the firstLine or hanging attributes. As well, if the startChars attribute is specified, then this value is ignored.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:start="720" w:end="2880" />`
	`</w:pPr>`
	
	This set of paragraph indentations specifies that this paragraph's text should be indented 720 twentieths of a point (one half inch) from the left text margin in this document, assuming this is a left to right paragraph. *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

startChars (Start	Specifies the indentation which shall be placed at the start of this paragraph – between

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Indentation in Character Units)	the left text margin for this paragraph and the left edge of that paragraph's content in a left to right paragraph, and the right text margin and the right edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§17.3.1.18) is specified for this paragraph, then this indent is used for the inside page edge - the right page edge for odd numbered pages and the left page edge for even numbered pages.
	
	This value is specified in hundredths of a character unit.
	
	If this attribute is omitted, its value shall be assumed to be zero.
	
	All other values for this element are relative to the leading text margin. Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. This value can be superseded for the first line only via use of the firstLine or hanging attributes. As well, if the start attribute is specified, then its value is ignored, and is superseded by this value.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:ind w:startChars="250" />`
	`</w:pPr>`
	
	This set of paragraph indentations specifies that this paragraph's text should be indented two and a half character units from the left text margin in this document, assuming this is a left to right paragraph. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ind) is located in §A.1. *end note*]

17.3.1.13 jc (Paragraph Alignment)

This element specifies the paragraph alignment which shall be applied to text in this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no alignment is applied to the paragraph.

[*Example*: Consider a paragraph which should be right justified to the right page side paragraph extents within a document. This constraint is specified in the following WordprocessingML content:

```
<w:pPr>
  <w:jc w:val="end" />
</w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The paragraph is now `right` justified on the page. *end example*]

Attributes	Description

val (Alignment Type)	Specifies the justification which should be applied to the parent object within a document.
	
	The possible values (see below) for this attribute are always specified with `left` specifying justification relative to the leading edge of the paragraph, and therefore change semantic between right-to-left and left-to-right documents.
	
	[*Example*: Consider the following WordprocessingML fragment for a paragraph in a document:
	
	`<w:pPr>`
	`  <w:jc w:val="end" />`
	`</w:pPr>`
	
	This paragraph is now right justified on the page for a left-to-right paragraph, left justified for a right-to-left paragraph. *end example*]
	
	The possible values for this attribute are defined by the ST_Jc simple type (§17.18.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Jc) is located in §A.1. *end note*]

17.3.1.14 keepLines (Keep All Lines On One Page)

This element specifies that all lines of this paragraph should be maintained on a single page whenever possible.

[*Note*: This means that if the contents of the current paragraph would normally span across two pages due to the placement of the paragraph's text, all lines in this paragraph should be moved onto the next page to ensure they are displayed together. If this is not possible because all lines in the paragraph would exceed a single page in any case, then lines in this paragraph should start on a new page, with page breaks as needed afterwards.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e., that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied. *end note*]

[*Example*: Consider a WordprocessingML document in which a code fragment (such as the schema fragments in this document) are defined such that they should never be broken across a page boundary in order to improve readability. This constraint would be specified using the following paragraph properties in WordprocessingML:

```
<w:pPr>
  <w:keepLines />
  …
</w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This setting ensures that the schema fragment is displayed on one page if possible. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

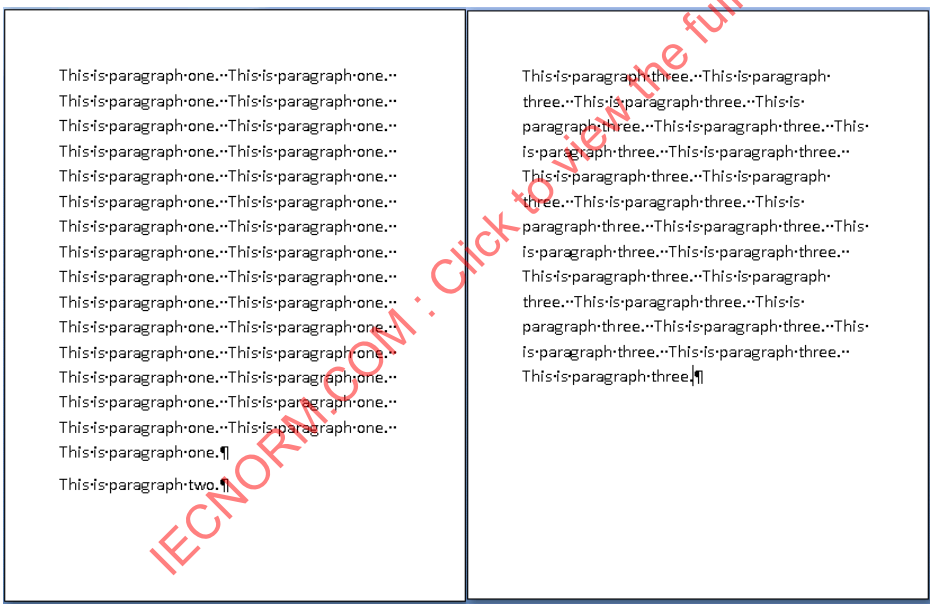
17.3.1.15 keepNext (Keep Paragraph With Next Paragraph)

This element specifies that the contents of this paragraph are at least partly rendered on the same page as the following paragraph whenever possible.

This means that if the contents of the current paragraph would normally be completely rendered on a different page than the following paragraph (because only one of the two paragraphs would fit on the remaining space on the first page), then both paragraphs shall be rendered on a single page. This property can be chained between multiple paragraphs to ensure that all paragraphs are rendered on a single page without any intervening page boundaries. If this is not possible the entire set of paragraphs that are grouped together using this property would exceed a single page in any case, then the set of "keep with next" paragraphs shall start on a new page, with page breaks as needed afterwards.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

[*Example*: Consider the following document with three paragraphs:



As shown above, the second paragraph and third paragraph are being rendered on two separate pages. However, a producer can specify that the second paragraph should not be displayed without any part of the third by setting the keepNext element as follows:

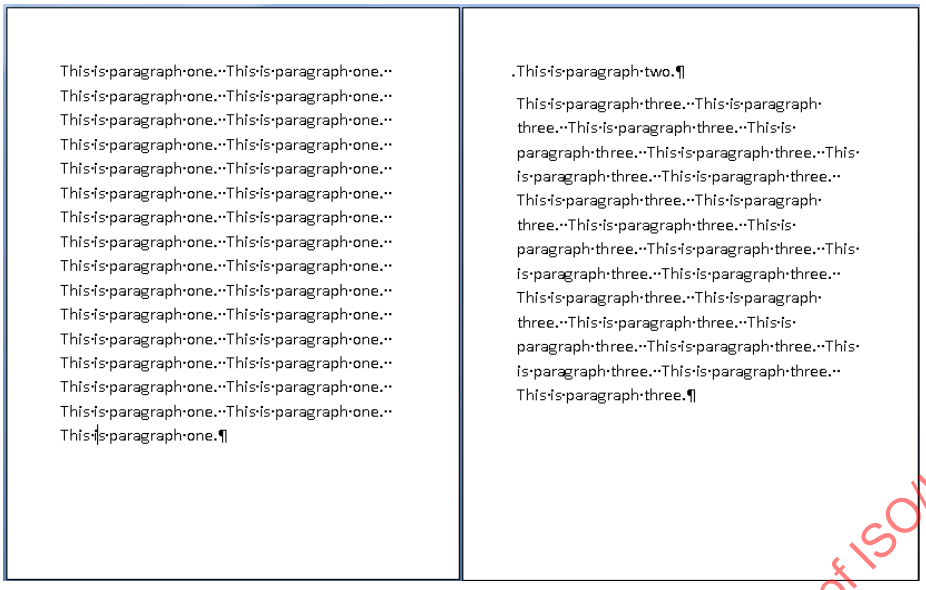
```
<w:pPr>
  <w:keepNext/>
  …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

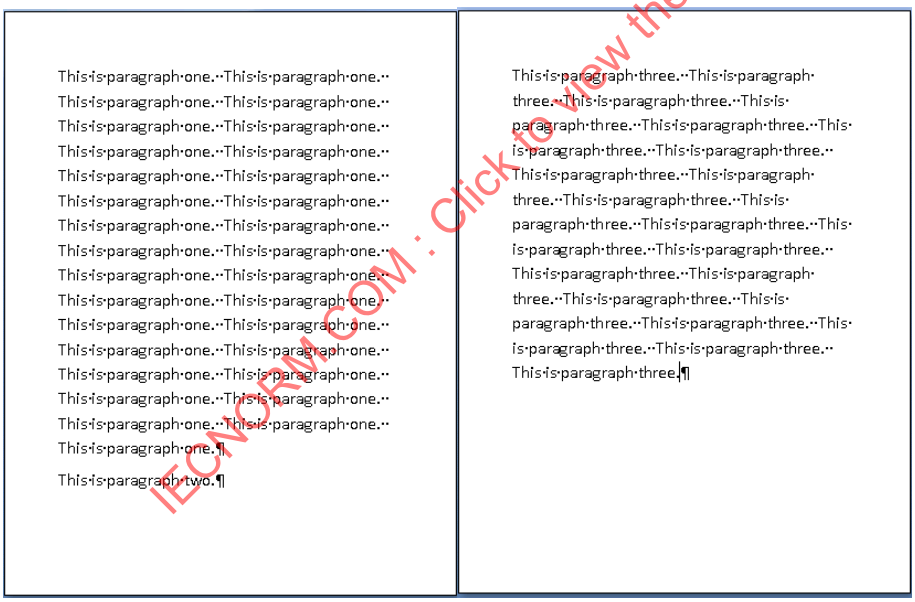
 ©ISO/IEC 2016 – All rights reserved

```
</w:pPr>
```

This would ensure that the second paragraph is displayed on the same page as the third paragraph:



However, if the first paragraph was also set to keepNext, then this set of grouped paragraphs would exceed a page, therefore the set of paragraphs would be grouped and start on the first page, resulting in the second paragraph reappearing on page one:



Since the paragraphs cannot all be put on one page, they all start on page one and flow as needed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.3.1.16 kinsoku (Use East Asian Typography Rules for First and Last Character per Line)

This element specifies whether East Asian typography and line-breaking rules shall be applied to text in this paragraph to determine which characters can begin and end each line. This property only applies to Simplified Chinese, Traditional Chinese, and Japanese text in this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall be applied to Simplified Chinese, Traditional Chinese, and Japanese text in this paragraph.

If these rules are set on the current paragraph, then the following rules are applied to the all first and last characters in the paragraph except the first and last character in the paragraph. By default, the following settings are used for kinsoku paragraphs:

Chinese (Simplified)

- Cannot start a line:
  !%),.:;>?]}¢ ¨ ° • ˇ ¯ — ‖ ’ ” … ‰ ′ ″ › ℃ ： 、 。 〃 〉 》 」 』 】 〕 〗 ″ ﹀ ﹂ ﹀ ﹀ ﹄ ） ｝ 〉 ！ ＂ ％ ＇ ） ， ． ： ； ？ ］ ｜ ｝ ～ ¢

  (Unicode character values: U+0021, U+0025, U+0029, U+002C, U+002E, U+003A, U+003B, U+003E, U+003F, U+005D, U+007D, U+00A2, U+00A8, U+00B0, U+00B7, U+02C7, U+02C9, U+2015, U+2016, U+2019, U+201D, U+2026, U+2030, U+2032, U+2033, U+203A, U+2103, U+2236, U+3001, U+3002, U+3003, U+3009, U+300B, U+300D, U+300F, U+3011, U+3015, U+3017, U+301E, U+FE36, U+FE3A, U+FE3E, U+FE40, U+FE44, U+FE5A, U+FE5C, U+FE5E, U+FF01, U+FF02, U+FF05, U+FF07, U+FF09, U+FF0C, U+FF0E, U+FF1A, U+FF1B, U+FF1F, U+FF3D, U+FF40, U+FF5C, U+FF5D, U+FF5E, and U+FFE0, respectively)

- Cannot end a line:
  $([{£¥·‘“ 〈 《 「 『 【 〔 〖 〝 （ ｛ ［ ＄ （. ［ ｛ ￡ ￥

  (Unicode character values: U+0024, U+0028, U+005B, U+007B, U+00A3, U+00A5, U+00B7, U+2018, U+201C, U+3008, U+300A, U+300C, U+300E, U+3010, U+3014, U+3016, U+301D, U+FE59, U+FE5B, U+FE5D, U+FF04, U+FF08, U+FF0E, U+FF3B, U+FF5B, U+FFE1, and U+FFE5, respectively)

Chinese (Traditional)

- Cannot start a line:
  !),.:;?]}¢ • － — ’ ” • … … ′ － 、 。 〉 》 」 』 】 ） ″ ： ｜｜ ｜ ﹀ ﹀ ﹂ ﹄ ﹀ ﹀ ﹄ ﹏ ， 、 ． ； ： ？ ！ ） ｝ 〉 ！ ） ， ． ： ； ？ ］ ｜ ｝ 、

  (Unicode character values: U+0021, U+0029, U+002C, U+002E, U+003A, U+003B, U+003F, U+005D, U+007D, U+00A2, U+00B7, U+2013, U+2014, U+2019, U+201D, U+2022, U+2025, U+2026, U+2027, U+2032, U+2574, U+3001, U+3002, U+3009, U+300B, U+300D, U+300F, U+3011, U+3015, U+301E,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

U+FE30, U+FE31, U+FE33, U+FE34, U+FE36, U+FE38, U+FE3A, U+FE3C, U+FE3E, U+FE40, U+FE42, U+FE44, U+FE4F, U+FE50, U+FE51, U+FE52, U+FE54, U+FE55, U+FE56, U+FE57, U+FE5A, U+FE5C, U+FE5E, U+FF01, U+FF09, U+FF0C, U+FF0E, U+FF1A, U+FF1B, U+FF1F, U+FF3D, U+FF5C, U+FF5D, and U+FF64, respectively)

- Cannot end a line:
  ([{£¥ ‘ “ 〈 《 「 『 【 〔 〝﹏﹏﹁﹃﹀﹀￢￡﹙ ｛ ﹛ （ ｛

  (Unicode character values: U+0028, U+005B, U+007B, U+00A3, U+00A5, U+2018, U+201C, U+2035, U+3008, U+300A, U+300C, U+300E, U+3010, U+3014, U+301D, U+FE35, U+FE37, U+FE39, U+FE3B, U+FE3D, U+FE3F, U+FE41, U+FE43, U+FE59, U+FE5B, U+FE5D, U+FF08, and U+FF5B, respectively)

Japanese

- Cannot start a line:
  !%),.:;?]}¢° ’ ” ‰′ ″ ℃、。々〉》」』】〕 ゝ ゞ ・ ヽ ヾ ！ ％ ） ， ． ： ； ？ ］ ｝ 。」、・ ゛ ゜ ¢

  (Unicode character values: U+0021, U+0025, U+0029, U+002C, U+002E, U+003A, U+003B, U+003F, U+005D, U+007D, U+00A2, U+00B0, U+2019, U+201D, U+2030, U+2032, U+2033, U+2103, U+3001, U+3002, U+3005, U+3009, U+300B, U+300D, U+300F, U+3011, U+3015, U+309B, U+309C, U+309D, U+309E, U+30FB, U+30FD, U+30FE, U+FF01, U+FF05, U+FF09, U+FF0C, U+FF0E, U+FF1A, U+FF1B, U+FF1F, U+FF3D, U+FF5D, U+FF61, U+FF63, U+FF64, U+FF65, U+FF9E, U+FF9F, and U+FFE0, respectively)

- Cannot end a line:
  $([\{£¥ ‘ “ 〈 《 「 『 【 〔 $ （ ［ ｛ 「 £ ￥

  (Unicode character values: U+0024, U+0028, U+005B, U+005C, U+007B, U+00A3, U+00A5, U+2018, U+201C, U+3008, U+300A, U+300C, U+300E, U+3010, U+3014, U+FF04, U+FF08, U+FF3B, U+FF5B, U+FF62, U+FFE1, and U+FFE5, respectively)

Korean

- Cannot start a line:
  !%),.:;?]}¢° ’ ” ‰′ ″ ℃〉》」』】〕！％），．：；？］｝¢

  (Unicode character values: U+0021, U+0025, U+0029, U+002C, U+002E, U+003A, U+003B, U+003F, U+005D, U+007D, U+00A2, U+00B0, U+2019, U+201D, U+2032, U+2033, U+2103, U+3009, U+300B, U+300D, U+300F, U+3011, U+3015, U+FF01, U+FF05, U+FF09, U+FF0C, U+FF0E, U+FF1A, U+FF1B, U+FF1F, U+FF3D, U+FF5D, and U+FFE0, respectively)

- Cannot end a line:
  $([\{£¥ ‘ “ 〈 《 「 『 【 〔 $ （ ［ ｛ £ ￥ ₩

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

(Unicode character values: U+0024, U+0028, U+005B, U+005C, U+007B, U+00A3, U+00A5, U+2018, U+201C, U+3008, U+300A, U+300C, U+300E, U+3010, U+3014, U+FF04, U+FF08, U+FF3B, U+FF5B, U+FFE1, U+FFE5, and U+FFE6, respectively)

If the strictFirstAndLastChars property (§17.15.1.82) is set in the Document Settings part, then the following settings supersede the defaults for Japanese:

- Cannot start a line:
  !%),.:;?]}¢° ’ ” ‰′ ″ ℃、。々〉》」』】〕ぁぃぅぇぉっゃゅょゎゕゖㇰㇱㇲㇳㇴㇵㇶㇷㇸㇹㇺㇻㇼㇽㇾㇿァィゥェォッャュョヮヵヶ・ーヽヾ！％），．：；？］｝。」、・ｱｲｳｴｵﾔﾕﾖﾂｰﾞﾟ¢

  (Unicode character values: U+0021, U+0025, U+0029, U+002C, U+002E, U+003A, U+003B, U+003F, U+005D, U+007D, U+00A2, U+00B0, U+2019, U+201D, U+2030, U+2032, U+2033, U+2103, U+3001, U+3002, U+3005, U+3009, U+300B, U+300D, U+300F, U+3011, U+3015, U+3041, U+3043, U+3045, U+3047, U+3049, U+3063, U+3083, U+3085, U+3087, U+308E, U+309B, U+309C, U+309D, U+309E, U+30A1, U+30A3, U+30A5, U+30A7, U+30A9, U+30C3, U+30E3, U+30E5, U+30E7, U+30EE, U+30F5, U+30F6, U+30FB, U+30FC, U+30FD, U+30FE, U+FF01, U+FF05, U+FF09, U+FF0C, U+FF0E, U+FF1A, U+FF1B, U+FF1F, U+FF3D, U+FF5D, U+FF61, U+FF63, U+FF64, U+FF65, U+FF67, U+FF68, U+FF69, U+FF6A, U+FF6B, U+FF6C, U+FF6D, U+FF6E, U+FF6F, U+FF70, U+FF9E, U+FF9F, and U+FFE0, respectively)

- Cannot end a line:
  $([\{£¥ ‘ “ 〈 《 「 『 【 〔 $ ([{「£￥

  (Unicode character values: U+0024, U+0028, U+005B, U+005C, U+007B, U+00A3, U+00A5, U+2018, U+201C, U+3008, U+300A, U+300C, U+300E, U+3010, U+3014, U+FF04, U+FF08, U+FF3B, U+FF5B, U+FF62, U+FFE1, and U+FFE5, respectively)

If the noLineBreaksBefore property (§17.15.1.59) is set in the Document Settings part, then the characters it specifies cannot begin a line for the specified language. If the noLineBreaksAfter property (§17.15.1.58) is set in the Document Settings part, then the characters it specifies cannot end a line for the specified language. In both cases, those settings shall supersede the defaults specified above.

[*Example*: Consider a document with a paragraph which should not use the kinsoku line breaking properties. This paragraph would define the following WordprocessingML:

```
<w:pPr>
  <w:kinsoku w:val="off" />
</w:pPr>
```

This paragraph would now be exempt from any kinsoku line breaking rules, and the characters specified above are allowed to begin and end lines as they normally would. *end example*]

This element’s content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.1.17 left (Left Paragraph Border)

This element specifies the border which shall be displayed on the left side of the page around the specified paragraph. This shall not change based on the paragraph direction.

To determine if any two adjoining paragraphs should have a left border which spans the full line height or not, the left border shall be drawn between the top border or between border at the top (whichever would be rendered for the current paragraph), and the bottom border or between border at the bottom (whichever would be rendered for the current paragraph).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no left border shall be applied.

[*Example*: Consider the following two paragraphs' WordprocessingML definition:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>First paragraph.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>Second paragraph.</w:t>
  </w:r>
</w:p>
```

Since the paragraph border set is identical between the two paragraphs, the paragraphs are connected by a between border. These paragraphs therefore draw the left border between the top and between borders for the first paragraph, and the between and bottom borders for the second paragraph. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.18 mirrorIndents (Use Left/Right Indents as Inside/Outside Indents)

This element specifies whether the paragraph indents should be interpreted as mirrored indents. When this element is present, the start indent shall become the inside indent (the one closest to the binding) and the end indent shall become the outside indent (the one furthest from the binding). [*Note*: This mirroring is typically used when the contents of the document are used to generate *signatures* – combinations of pages which are then placed in a binding. When signatures are printed in a left-to-right document, the first, third, etc. pages are printed on the left side of the combined sheet, and the second, fourth, etc. are printed on its right side, then bound and folded. For a right-to-left document, the first, third, etc. pages are printed on the right side of the combined sheet, and the second, fourth, etc. are printed on its left side. *end note*]

If the mirrorIndents property is specified for this paragraph, then the inside page edge is the end page edge for odd numbered pages and the start page edge for even numbered pages. Conversely, the outside page edge is the start page edge for odd numbered pages and the end page edge for even numbered pages. Odd and even numbering in the text above refers to the ordinal position of the page in the paginated document, not to the page number which may appear on each page.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

[*Example*: Consider a paragraph on the first page of a document which should have a one inch indentation from the text margins on the inside edge when the resulting document is printed and bound. This means that the paragraph has a one inch right border if it is on an odd numbered page, and a one inch left border if it is on an even numbered page. This set of indentations is specified using the following WordprocessingML:

```
<w:pPr>
  <w:ind w:start="1440" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:mirrorIndents />
  </w:pPr>
```

This set of indentation properties specifies that a 1440 twip indentation should be provided on the leading side of the text margins for this paragraph. However, since the mirrorIndents property is set, the start indent is really the inside indent, and if this paragraph is on page one, must result in a one inch right indent from the text margin. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.19 numPr (Numbering Definition Instance Reference)

This element specifies that the current paragraph uses numbering information that is defined by a particular *numbering definition instance*.

The presence of this element specifies that the paragraph inherits the properties specified by the numbering definition in the num element (§17.9.15) at the level specified by the level specified in the lvl element (§17.9.6) and shall have an associated number positioned before the beginning of the text flow in this paragraph. When this element appears as part of the paragraph formatting for a paragraph style, then any numbering level defined using the ilvl element shall be ignored, and the pStyle element (§17.9.23) on the associated abstract numbering definition shall be used instead.

[*Example*: Consider a paragraph in a document which should be associated with level 4 of a numbering definition with ID 0. Associating the paragraph with this numbering definition would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:numPr>
    <w:ilvl w:val="4" />
    <w:numId w:val="0" />
  </w:numPr>
</w:pPr>
```

The numPr element specifies that this paragraph must contain numbering information, and its children specify that the numbering definition for that numbering information must have a numId of 0 and an ilvl of 4 within that numbering definition. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumPr) is located in §A.1. *end note*]

17.3.1.20 outlineLvl (Associated Outline Level)

This element specifies the *outline level* which shall be associated with the current paragraph in the document. The *outline level* specifies an integer which defines the level of the associated text. This level shall not affect the appearance of the text in the document, but shall be used to calculate the TOC field (§17.16.5.68) if the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

appropriate field switches have been set, and can be used by consumers to provide additional application behavior.

The outline level of text in the document (specified using the val attribute) can be from 0 to 9, where 9 specifically indicates that there is no outline level specifically applied to this paragraph. If this element is omitted, then the outline level of the content is assumed to be 9 (no level).

[*Example*: Consider a paragraph in a document which has outline level 1 applied to it. This paragraph would specify the following WordprocessingML:

```
<w:pPr>
  <w:outlineLvl w:val="0" />
</w:pPr>
```

This paragraph is now of outline level 1, and if a table of contents field is inserted that utilizes outlines levels, the text in this paragraph is at level one in the TOC. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: <… w:val="1512645511" /> The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.3.1.21 overflowPunct (Allow Punctuation to Extend Past Text Extents)

This element specifies that the text in this paragraph shall be allowed to extend one character beyond the extents applied by any indents/margins when the character that extends past those extents is a punctuation character.

Omitting this element sets its value to `true`.

[*Example*: Consider a WordprocessingML document with the following string at the end of a line:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
"This is some text in quotation marks"
```

Typically, if the text extents would normally fall between the letter s and the closing quotation mark, the quotation mark would be allowed to extend past the end of the line by one character even though the punctuation is not part of the word marks (since the omission of overflowPunct is equivalent to setting its val attribute to true).

However, if this behavior should not be applied to this paragraph, a producer can specify this by setting the property in the WordprocessingML:

```
<w:pPr>
  <w:overflowPunct w:val="0" />
</w:pPr>
```

The line would now break after the letter s, regardless of the fact that the next character is a quotation mark. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.22 p (Paragraph)

This element specifies a paragraph of content in the document.

The contents of a paragraph in a WordprocessingML document shall consist of any combination of the following four kinds of content:

- Paragraph properties
- Annotations (bookmarks, comments, revisions)
- Custom markup
- Run level content (fields, hyperlinks, runs)

[*Example*: Consider a basic WordprocessingML document with a single paragraph. This paragraph would be expressed as follows:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Text</w:t>
      </w:r>
      <w:fldSimple w:instr="AUTHOR">
        <w:r>
          <w:t>Author Name</w:t>
        </w:r>
      </w:fldSimple>
    </w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </w:body>
  </w:document>
```

The p element is the container for all of the content in the paragraph, which in this example includes both a text run and a simple field. *end example*]

Attributes	Description
rsidDel (Revision Identifier for Paragraph Deletion)	Specifies an identifier used to track the editing session when the paragraph was deleted from the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidP (Revision Identifier for Paragraph Properties)	This attribute specifies an identifier used to track the editing session when the paragraph's properties were last modified in this document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Revision Identifier for Paragraph)	This attribute specifies an identifier used to track the editing session when the paragraph was added to the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
rsidRDefault (Default Revision Identifier for Runs)	This attribute specifies an identifier used for all runs in this paragraph which do not explicitly declare an rsidR attribute. This attribute allows consumers to optimize the locations where rsid* values are written in this document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidRPr (Revision Identifier for Paragraph Glyph Formatting)	This attribute specifies an identifier used to track the editing session when the glyph character representing the paragraph mark was last modified in the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_P) is located in §A.1. *end note*]

17.3.1.23 pageBreakBefore (Start Paragraph on Next Page)

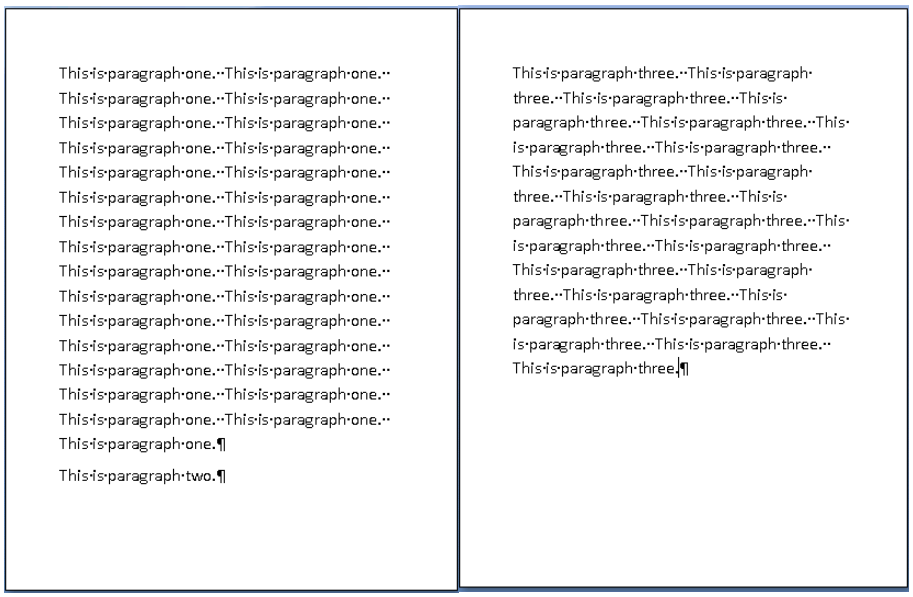
This element specifies that the contents of this paragraph are rendered on the start of a new page.

This means that if the contents of the current paragraph would normally be rendered on the middle of a page in the host document, then the paragraph shall be rendered on a new page as if the paragraph was preceded by a page break in the WordprocessingML contents of the document. This property supersedes any use of the keepNext property, so that if any paragraph wishes to be on the same page as this paragraph, they are still be separated by a page break.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

[*Example*: Consider the following document with three paragraphs:

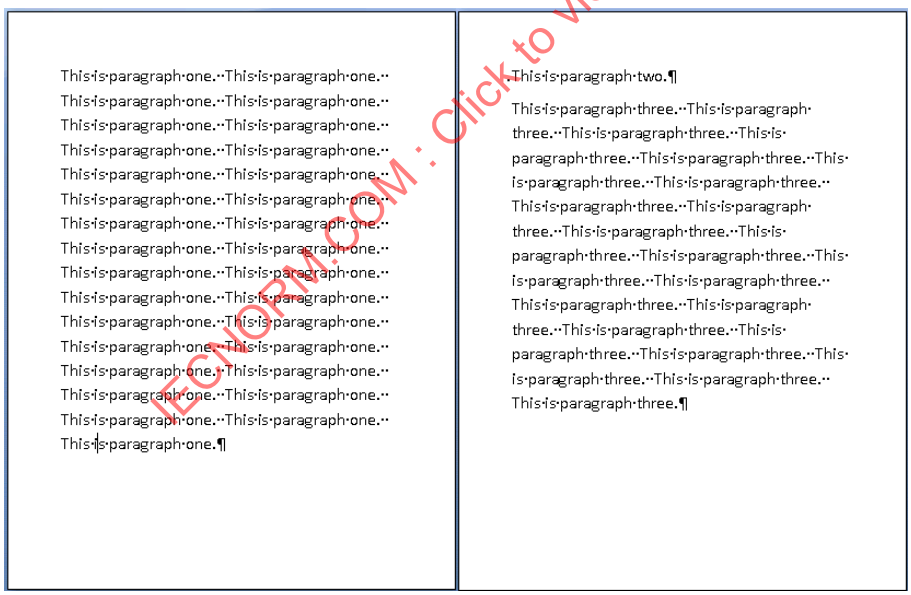
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



As shown above, the second paragraph is rendered at the bottom of page one. However, a producer can specify that the second paragraph should be displayed at the top of a new page by setting the pageBreakBefore element as follows:

```
<w:pPr>
  <w:pageBreakBefore/>
</w:pPr>
```

This would ensure that the second paragraph is displayed on a new page:



Since the paragraph is specified to start on a new page, it begins page two even though it could have fit on page one. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.24 pBdr (Paragraph Borders)

This element specifies the borders for the parent paragraph. Each child element shall specify a specific kind of border (left, right, bottom, top, and between).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no paragraph borders shall be applied.

[*Example*: Consider a pairing of paragraphs which have a three point red border around them, and a six point border between them. These paragraphs would each have the following set of paragraph borders:

```
<w:pBdr>
  <w:top w:val="single" w:sz="24" w:space="1" w:color="FF0000" />
  <w:left w:val="single" w:sz="24" w:space="4" w:color="FF0000" />
  <w:bottom w:val="single" w:sz="24" w:space="1" w:color="FF0000" />
  <w:right w:val="single" w:sz="24" w:space="4" w:color="FF0000" />
  <w:between w:val="single" w:sz="48" w:space="1" w:color="4D5D2C" />
</w:pBdr>
```

The resulting paragraphs have identical pBdr values, therefore they would use the top, left, bottom, and right borders around them as a units, and the between border between each other. This matching heuristic is further discussed in the child elements of the pBdr element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PBdr) is located in §A.1. *end note*]

17.3.1.25 pPr (Previous Paragraph Properties)

This element specifies a set of paragraph properties which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

[*Example*: Consider a paragraph which should have a set of paragraph formatting properties that were added with revision tracking turned on. This set of revised properties is specified in the paragraph properties as follows:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:bottom w:val="single" w:sz="8" w:space="4" w:color="4F81BD" />
    </w:pBdr>
    <w:pPrChange w:author="user1" … >
      <w:pPr>
        <w:spacing w:after="300" />
        <w:contextualSpacing />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </w:pPr>
      </w:pPrChange>
    </w:pPr>
  </w:p>
```

The pPr element under pPrChange specifies the properties which are applied to the current paragraph with revision tracking turned on - in this case, spacing after the paragraph using the spacing element (§17.3.1.33), and that spacing should be ignored for paragraphs above/below of the same style using the contextualSpacing element (§17.3.1.9). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrBase) is located in §A.1. *end note*]

17.3.1.26 pPr (Paragraph Properties)

This element specifies a set of paragraph properties which shall be applied to the contents of the parent paragraph after all style/numbering/table properties have been applied to the text. These properties are defined as *direct formatting*, since they are directly applied to the paragraph and supersede any formatting from styles.

[*Example*: Consider a paragraph which should have a set of paragraph formatting properties. This set of properties is specified in the paragraph properties as follows:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:bottom w:val="single" w:sz="8" w:space="4" w:color="4F81BD" />
    </w:pBdr>
    <w:spacing w:after="300" />
    <w:contextualSpacing />
  </w:pPr>
</w:p>
```

The pPr element specifies the properties which are applied to the current paragraph - in this case, a bottom paragraph border using the bottom element (§17.3.1.7), spacing after the paragraph using the spacing element (§17.3.1.33), and that spacing should be ignored for paragraphs above/below of the same style using the contextualSpacing element (§17.3.1.9). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPr) is located in §A.1. *end note*]

17.3.1.27 pStyle (Referenced Paragraph Style)

This element specifies the style ID of the paragraph style which shall be used to format the contents of this paragraph.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Table styles
- Numbering styles
- Paragraph styles (this element)
- Character styles
- Direct Formatting

This means that all properties specified in the style element (§17.7.4.17) with a styleId which corresponds to the value in this element's val attribute are applied to the paragraph at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no paragraph style shall be applied to the current paragraph. As well, this property is ignored if the paragraph properties are part of a paragraph style.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:pPr>
  <w:pStyle w:val="TestParagraphStyle" />
  <w:ind w:start="1440" />
</w:pPr>
```

This paragraph specifies that it inherits all of the paragraph properties specified by the paragraph style with a styleId of `TestParagraphStyle`, which then has any indentation properties overridden with a start indentation of `1440` twentieths of a point, and no indentation for any other value. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.3.1.28 right (Right Paragraph Border)

This element specifies the border which shall be displayed on the right side of the page around the specified paragraph. This shall not change based on the paragraph direction.

To determine if any two adjoining paragraphs should have a right border which spans the full line height or not, the right border shall be drawn between the top border or between border at the top (whichever would be rendered for the current paragraph), and the bottom border or between border at the bottom (whichever would be rendered for the current paragraph).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no right border shall be applied.

[*Example*: Consider the following two paragraphs' WordprocessingML definition:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>First paragraph.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>Second paragraph.</w:t>
  </w:r>
</w:p>
```

Since the paragraph border set is identical between the two paragraphs, the paragraphs are connected by a between border. These paragraphs therefore draw the right border between the top and between borders for the first paragraph, and the between and bottom borders for the second paragraph. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.29 rPr (Run Properties for the Paragraph Mark)

This element specifies the set of run properties applied to the glyph used to represent the physical location of the paragraph mark for this paragraph. This paragraph mark, being a physical character in the document, can be formatted, and therefore shall be capable of representing this formatting like any other character in the document.

If this element is not present, the paragraph mark is unformatted, as with any other run of text.

[*Example*: Consider a run of text displayed as follows, including a display format using the pilcrow sign ¶ for the paragraph mark glyph:

```
This is some text and the paragraph mark.¶
```

If we format the display formatting for the paragraph mark by making it red and giving it a 72 point font size, then the WordprocessingML must reflect this formatting on the paragraph as follows:

```
<w:pPr>
  <w:rPr>
    <w:color w:val="FF0000" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <w:sz w:val="144" />
    </w:rPr>
  </w:pPr>
```

The paragraph glyph's formatting is stored in the rPr element under the paragraph properties, since there is no run saved for the paragraph mark itself. *end example*]

The W3C XML Schema definition of this element's content model (CT_ParaRPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

17.3.1.30 rPr (Previous Run Properties for the Paragraph Mark)

This element specifies a set of run properties applied to the glyph used to represent the physical location of the paragraph mark for this paragraph which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

[*Example*: Consider a run which has a set of run formatting properties that were added with revision tracking turned on. This set of revised properties is specified in the run properties as follows:

```
<w:p>
  <w:pPr>
    <w:rPr>
      <w:b />
      <w:imprint />
      <w:lang w:val="en-CA" />
      <w:rPrChange w:id="1" w:author="user1">
        <w:rPr>
          <w:i />
          <w:dstrike w:val="false" />
        </w:rPr>
      </w:rPrChange>
    </w:rPr>
  </w:pPr>
</w:p>
```

The rPr element under rPrChange specifies the properties which are applied to the run representing the paragraph mark before the revision tracking was turned on - in this case, italics using the i element (§17.3.2.16), and that any double strikethrough which was applied based on the style hierarchy must be turned off using the dstrike element (§17.3.2.9). *end example*]

The W3C XML Schema definition of this element's content model (CT_ParaRPrOriginal) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.1.31 shd (Paragraph Shading)

This element specifies the shading applied to the contents of the paragraph.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no paragraph shading shall be applied.

[*Example*: Consider a paragraph which must have a background consisting of a theme color `accent3` with a theme color `accent6` overlaid using a 20% fill pattern. This requirement is specified using the following WordprocessingML:

```
<w:pPr>
  <w:shd w:val="pct20" w:themeColor="accent6" w:themeFill="accent3" />
</w:pPr>
```

The resulting paragraph uses the background color `accent3` under the foreground pattern color `accent6` as specified by the `pct20` pattern mask. *end example*]

This element's content model is defined by the common shading properties definition in §17.3.5.

17.3.1.32 snapToGrid (Use Document Grid Settings for Inter-Line Paragraph Spacing)

This element specifies whether the current paragraph should use the document grid lines per page settings defined in the docGrid element (§17.6.5) when laying out the contents in the paragraph. This setting determines whether the additional line pitch specified in the document grid shall be added to each line in this paragraph as specified by the document grid.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall use the document grid to lay out text when a document grid is defined for this document.

[*Example*: Consider two single-spaced paragraphs in a section with a document grid set to allow 15 lines per page. This document grid would effectively specifies that an additional line pitch of 45.6 points must be added to each line in order to ensure that the resulting page contains only 15 lines of text.

If this property is set on the first paragraph, but turned off on the second paragraph, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:pPr>
    <w:snapToGrid w:val="off" />
  </w:pPr>
  …
</w:p>
<w:p>
  …
</w:p>
```

The resulting document must have 45.6 points of additional line pitch added to each line in paragraph two, but zero lines of additional line pitch added to each line in paragraph one, since the snapToGrid property is turned off. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.33 spacing (Spacing Between Lines and Above/Below Paragraph)

This element specifies the inter-line and inter-paragraph spacing which shall be applied to the contents of this paragraph when it is displayed by a consumer.

If this element is omitted for a given paragraph, the values of the settings represented by each of its attributes is determined by the setting previously set at any level of the style hierarchy (i.e., that previous setting remains unchanged). If a setting is not previously specified in the style hierarchy, then its value is as described by the corresponding attribute below.

[*Example*: Consider the following WordprocessingML paragraph:

```
<w:pPr>
  <w:spacing w:after="200" w:line="276" w:lineRule="auto" />
</w:pPr>
```

This paragraph specifies that it must have at least 200 twentieths of a point after the last line in each paragraph, and that the spacing in each line should be automatically calculated based on a 1.15 times (276 divided by 240) the normal single spacing calculation. *end example*]

When determining the spacing between any two paragraphs, a consumer shall use the maximum of the inter-line spacing in each paragraph, the spacing after the first paragraph and the spacing before the second paragraph to determine the net spacing between the paragraphs.

[*Example*: Consider two consecutive single-spaced paragraphs in a document, the first of which specifies spacing below of 12 points, the second of which specifies spacing above of 4 points. These constraints are expressed using the following WordprocessingML:

```
<w:p>
  <w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <w:spacing w:after="240" />
    </w:pPr>
    …
  </w:p>
  <w:p>
    <w:pPr>
      <w:spacing w:before="80" />
    </w:pPr>
    …
  </w:p>
```

The resulting spacing between the first and second paragraph is 12 points, since that is the largest spacing requested between the two paragraphs. *end example*]

Attributes	Description

after (Spacing Below Paragraph)	Specifies the spacing that should be added after the last line in this paragraph in the document in absolute units.
	
	If this attribute is omitted on a given paragraph, the value of the setting it represents is the value previously set in the style hierarchy. If this setting is never specified in the style hierarchy, then the paragraph shall have no spacing applied below its contents.
	
	If the afterLines attribute or the afterAutoSpacing attribute is also specified, then this attribute value is ignored.
	
	[*Example*: Consider the following WordprocessingML paragraph:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:spacing w:after="240" />`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph must have a minimum spacing below its final lines of 240 twentieths of a point, although the actual spacing can be determined by the inter-line spacing or the spacing above the following paragraph, if either are greater. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

afterAutospacing (Automatically Determine Spacing Below Paragraph)	Specifies whether a consumer shall automatically determine the spacing after this paragraph based on its contents. This automatic spacing shall match the spacing which would be applied to the paragraph in an HTML document where no explicit spacing before/after is specified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	If this attribute is specified, then any value in after or afterLines is ignored, and the spacing is automatically determined by the consumer.
	
	If this attribute is omitted on a given paragraph, the value of the setting it represents is the value previously set in the style hierarchy. If this setting is never specified in the style hierarchy, then automatic spacing is turned off.
	
	[*Example*: Consider a paragraph in a document whose spacing below must automatically be determined by the consumer based on the paragraph's contents. This constraint would be specified by the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:spacing … w:afterAutospacing="1" />`
	`</w:pPr>`
	
	The resulting paragraph must have the spacing below its last line determined automatically by the consumer to match an HTML document as specified. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

afterLines (Spacing Below Paragraph in Line Units)	Specifies the spacing that should be added after the last line in this paragraph in the document in line units.
	
	The value of this attribute is specified in one hundredths of a line.
	
	If the afterAutoSpacing attribute is also specified, then this attribute value is ignored. If this setting is never specified in the style hierarchy, then its value shall be zero.
	
	[*Example*: Consider the following WordprocessingML paragraph:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:spacing w:afterLines="300" />`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph must have a minimum spacing below its final lines of 3 lines, although the actual spacing can be determined by the inter-line spacing or the spacing above the following paragraph, if either are greater. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

before (Spacing Above Paragraph)	Specifies the spacing that should be added above the first line in this paragraph in the document in absolute units. If this attribute is omitted on a given paragraph, the value of the setting it represents is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	the value previously set in the style hierarchy. If this setting is never specified in the style hierarchy, then the paragraph shall have no spacing applied above its contents.
	
	If the beforeLines attribute or the beforeAutoSpacing attribute is also specified, then this attribute value is ignored.
	
	[*Example*: Consider the following WordprocessingML paragraph:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:spacing w:before="80" />`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph must have a minimum spacing above its first line of 80 twentieths of a point, although the actual spacing can be determined by the inter-line spacing or the spacing below the last line in the preceding paragraph, if either are greater. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

beforeAutospacing (Automatically Determine Spacing Above Paragraph)	Specifies whether a consumer shall automatically determine the spacing before this paragraph based on its contents.
	
	This automatic spacing shall match the spacing which would be applied to the paragraph in an HTML document where no explicit spacing before/after is specified.
	
	If this attribute is specified, then any value in before or beforeLines is ignored, and the spacing is automatically determined by the consumer.
	
	If this attribute is omitted on a given paragraph, the value of the setting it represents is the value previously set it the style hierarchy. If this setting is never specified in the style hierarchy, then automatic spacing is turned off.
	
	[*Example*: Consider a paragraph in a document whose spacing above must automatically be determined by the consumer based on the paragraph's contents. This constraint would be specified by the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:spacing … w:beforeAutospacing="1" />`
	`</w:pPr>`
	
	The resulting paragraph must have the spacing above its first line determined automatically by the consumer to match an HTML document as specified. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

beforeLines (Spacing Above Paragraph in Line Units)	Specifies the spacing that should be added before the first line in this paragraph in the document in line units.
	
	The value of this attribute is specified in one hundredths of a line.
	
	If the beforeAutoSpacing attribute is also specified, then this attribute value is ignored. If this setting is never specified in the style hierarchy, then its value shall be zero.
	
	[*Example*: Consider the following WordprocessingML paragraph:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:spacing w:beforeLines="100" />`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph must have a minimum spacing above its first line of 1 line, although the actual spacing can be determined by the inter-line spacing or the spacing below the preceding paragraph, if either are greater. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

line (Spacing Between Lines in Paragraph)	This attribute specifies the amount of vertical spacing between lines of text within this paragraph. If this attribute is omitted on a given paragraph, the value of the setting it represents is the value previously set in the style hierarchy. If this setting is never specified in the style hierarchy, then single (no additional) line spacing shall be applied to lines within this paragraph. If the value of the lineRule attribute is either `atLeast` or `exact`, then the value of this attribute shall be interpreted as twentieths of a point. When the value of the lineRule attribute is either `atLeast` or `exact`, the text shall be positioned as follows within that line height: • When the line height is too small, the text shall be positioned at the bottom of the line (i.e. clipped from the top down) • When the line height is too large, the text shall be centered in the available space. If the value of the lineRule attribute is `auto`, then the value of the line attribute shall be interpreted as 240ths of a line, in the manner described by the simple type's values. [*Example*: Consider the following WordprocessingML paragraph which should have an inter-line spacing of 1.15 times the line height. This constraint would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:spacing w:line="276" w:lineRule="auto" /> </w:pPr> The lineRule attribute value of auto specifies that the value of the line attribute is to be interpreted in 240ths of a single line height, which means that the net spacing is 276/240ths of a line or 1.15 lines tall. *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).
lineRule (Spacing Between Lines)	Specifies how the spacing between lines is calculated as stored in the line attribute. If this attribute is omitted, then it shall be assumed to be of a value auto if a line attribute value is present. If the value of this attribute is either atLeast or exactly, then the value of the line attribute shall be interpreted as twentieths of a point, in the manner described by the simple type's values. If the value of this attribute is auto, then the value of the line attribute shall be interpreted as 240ths of a line, in the manner described by the simple type's values. [*Example*: Consider the following WordprocessingML paragraph which should have an inter-line spacing of 1.15 times the line height. This constraint would be specified using the following WordprocessingML: <w:pPr> <w:spacing w:line="276" w:lineRule="auto" /> </w:pPr> The lineRule attribute value of auto specifies that the value of the line attribute is to be interpreted in 240ths of a single line height. *end example*] The possible values for this attribute are defined by the ST_LineSpacingRule simple type (§17.18.48).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Spacing) is located in §A.1. *end note*]

17.3.1.34 suppressAutoHyphens (Suppress Hyphenation for Paragraph)

This element specifies whether any hyphenation shall be performed on this paragraph by the consumer when requested using the autoHyphenation element (§17.15.1.10) in the document's settings. This element specifies

©ISO/IEC 2016 – All rights reserved

whether the current paragraph should be exempted from any hyphenation which is applied by the consumer on this document.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the default hyphenation settings for the document, as specified in the autoHyphenation element, shall apply to the contents of this paragraph.

[*Example*: Consider a document which must be hyphenated automatically by a consumer, since it has the autoHyphenation element set to `true` in its document settings. If this paragraph should be exempted from that hyphenation pass, this requirement would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:suppressAutoHyphens />
</w:pPr>
```

The paragraph would then be exempted from hyphenation by a consumer at display time, regardless of the hyphenation settings for the document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.35 suppressLineNumbers (Suppress Line Numbers for Paragraph)

This element specifies whether line numbers shall be calculated for lines in this paragraph by the consumer when line numbering is requested using the lnNumType element (§17.6.8) in the paragraph's parent section settings. This element specifies whether the current paragraph's lines should be exempted from line numbering which is applied by the consumer on this document, not just suppressing the display of the numbering, but removing these lines from the line numbering calculation.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the default line number settings for the section, as specified in the lnNumType element shall apply to each line of this paragraph.

[*Example*: Consider a document with three paragraphs, each of which are displayed on five lines , all contained in a section which has the lnNumType element specified. If the second paragraph should be exempted from that line numbering, this requirement would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:suppressLineNumbers />
</w:pPr>
```

The paragraph would then be exempted from line by a consumer at display time, which would result in paragraph one using line numbers one through five, the second paragraph having no line numbers, and the third paragraph using line numbers six through ten. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.36 suppressOverlap (Prevent Text Frames From Overlapping)

This element specifies whether a text frame which intersects another text frame at display time shall be allowed to overlap the contents of the other text frame. If a text frame cannot overlap other text frames, it shall be repositioned when displayed to prevent this overlap as needed.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then overlap shall be allowed between a text frame which intersects another text frame displayed at the same location.

[*Example*: Consider a document with two text frames which are allowed to overlap each other. If the second text frame should overlap the contents of another text frame, that constraint would be specified via the following WordprocessingML:

```
<w:p>
  …
</w:p>
<w:p>
  <w:pPr>
    <w:framePr … />
    <w:suppressOverlap />
  </w:pPr>
  …
</w:p>
```

The resulting text frame with the suppressOverlap property specified would never overlap any intersecting text frames. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.37 tab (Custom Tab Stop)

This element specifies a single custom tab stop defined within a set of paragraph properties in a document. A tab stop location shall always be measured relative to the leading edge of the paragraph in which it is used (that is, the left edge for a left-to-right paragraph, and the right edge for a right-to-left paragraph).

[*Example*: Consider a custom tab stops at 1.5" in a WordprocessingML document. This tab stop would be contained within a tab element defining the tab stop as follows:

```
<w:tab w:val="start" w:pos="2160" />
```

The tab element specifies all of the properties for the custom tab stop for the current paragraph property set. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
leader (Tab Leader Character)	Specifies the character which shall be used to fill in the space created by a tab which ends at this custom tab stop. This character shall be repeated as required to completely fill the tab spacing generated by the tab character. If this attribute is omitted, then no tab leader character shall be used. [*Example*: Consider a tab stop which should be preceded by a sequence of underscore characters, as follows: _____Text at the tab stop This tab stop would have a leader attribute value of `underscore`, indicating that the tab stop must be preceded by underscore characters as needed to fill the tab spacing. *end example*] The possible values for this attribute are defined by the ST_TabTlc simple type (§17.18.85).
pos (Tab Stop Position)	Specifies the position of the current custom tab stop with respect to the current page margins. Negative values are permitted and move the tab stop into the current page margin the specified amount. [*Example*: Consider a custom tab stops at 1.5" in a WordprocessingML document. This tab stop would be contained within a tab element defining the tab stop as follows: `<w:tab w:val="start" w:pos="2160" />` The pos attribute specifies that this custom tab stop must be located `2160` points (1.5 inches) inside the starting text margin. *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).
val (Tab Stop Type)	Specifies the style of custom tab stop, which determines the behavior of the tab stop and the alignment which shall be applied to text entered at the current custom tab stop. The value of `clear` is unique and specifies that this tab stop shall be removed when the document is next edited by a consumer which supports rendering the document contents. [*Example*: Consider a custom tab stops at 1.5" in a WordprocessingML document. This tab stop would be contained within a tab element defining the tab stop as follows: `<w:tab w:val="start" w:pos="2160" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The val attribute specifies that this custom tab stop must align all text entered at its location to its left. *end example*] The possible values for this attribute are defined by the ST_TabJc simple type (§17.18.84).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TabStop) is located in §A.1. *end note*]

17.3.1.38 tabs (Set of Custom Tab Stops)

This element specifies a sequence of custom tab stops which shall be used for any tab characters in the current paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no custom tab stops shall be used for this paragraph.

As well, this property is additive - tab stops at each level in the style hierarchy are added to each other to determine the full set of tab stops for the paragraph. A hanging indent specified via the hanging attribute on the ind element (§17.3.1.12) shall also always implicitly create a custom tab stop at its location.

[*Example*: Consider a paragraph which contains two custom tab stops at 1.5" and 3.5", respectively. These two tab stops would be contained within a tabs element defining the set of tab stops of the paragraph as follows:

```
<w:pPr>
  <w:tabs>
    <w:tab w:val="start" w:pos="2160" />
    <w:tab w:val="start" w:pos="5040" />
  </w:tabs>
</w:pPr>
```

The tabs element specifies all of the customized tab stops for the current paragraph. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tabs) is located in §A.1. *end note*]

17.3.1.39 textAlignment (Vertical Character Alignment on Line)

This element specifies the vertical alignment of all text on each line displayed within a paragraph. If the line height (before any added spacing) is larger than one or more characters on the line, all characters are aligned to each other as specified by this element.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

hierarchy, then the vertical alignment of all characters on the line shall be automatically determined by the consumer.

[*Example*: Consider a paragraph of text of different font sizes, as follows:

If the text on this paragraph must be aligned based on the top point of the maximum character height, that requirement would be specified as follows in the WordprocessingML:

```
<w:pPr>
  <w:textAlignment w:val="top" />
</w:pPr>
```

The resulting text would be top aligned, as follows:

The characters are all aligned to the maximum character extent on the line. *end example*]

Attributes	Description

val (Vertical Character Alignment Position)	Specifies the style of vertical alignment which shall be used to align the characters on each line in the current paragraph.
	
	[*Example*: Consider a paragraph of text of different font sizes which must be aligned based on the baseline point of each character in each line. This requirement would be specified as follows in the WordprocessingML:
	
	`<w:pPr>`
	`  <w:textAlignment w:val="baseline" />`
	`</w:pPr>`
	
	The resulting text would be aligned to the baseline for each character on the line. *end example*]
	
	The possible values for this attribute are defined by the ST_TextAlignment simple type (§17.18.91).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextAlignment) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.1.40 textboxTightWrap (Allow Surrounding Paragraphs to Tight Wrap to Text Box Contents)

This element specifies whether, for paragraphs in a text box, the surrounding text shall be allowed to overlap with the empty text box boundaries and tight wrap to the extents of the text within the text box.

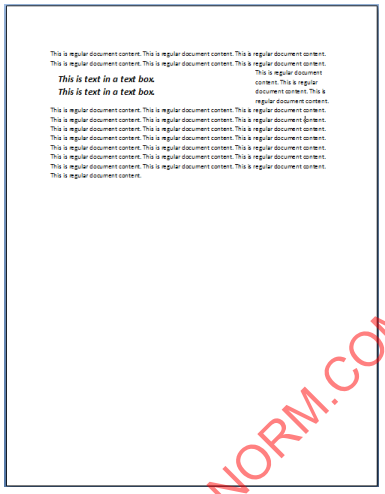
This element shall only be read for paragraphs which are contained within a text box (have a txbxContent ancestor), ignored otherwise.

If the parent text box does not meet the following three criteria, then this property has no effect:

- The text box wrapping shall be set to `tight`
- The text box border shall not be set
- The text box shading shall not be set

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then paragraphs in a text box have no tight wrapping overrides, and text shall wrap to the extents of the text box.

[*Example*: Consider a document with a tight wrapped text box which extends two-thirds of the way across the page, as follows:

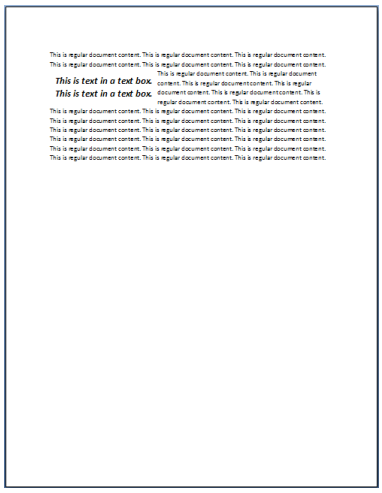


The surrounding text is tightly wrapped to the extents of the text box. If the consumer must tight wrap to the extents of the text, that requirement would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:textboxTightWrap w:val="allLines" />
</w:pPr>
```

This would result in the following display of the content:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



The resulting paragraphs within the textbox use the textboxTightWrap element to specify that text should be tightly wrapped to the paragraph's extents. *end example*]

Attributes	Description

val (Lines to Tight Wrap to Paragraph Extents)	Specifies the lines in the parent paragraph which shall allow the text to be tight wrapped to the paragraph (and not the text box) extents when displaying the document.
	
	[*Example*: Consider a paragraph in a text box which meets the criteria specified above which must allow wrapping to the text extents on its first line only. That requirement would be specified using the following WordprocessingML:
	
	`<w:pPr>`
	`  <w:textboxTightWrap w:val="firstLineOnly" />`
	`</w:pPr>`
	
	The resulting paragraph would allow text to tightly wrap to the contents of its first line only. All other lines would wrap to the text box's extents. *end example*]
	
	The possible values for this attribute are defined by the ST_TextboxTightWrap simple type (§17.18.92).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextboxTightWrap) is located in §A.1. *end note*]

17.3.1.41 textDirection (Paragraph Text Flow Direction)

This element specifies the direction of the text flow for this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall inherit the text flow settings from the parent section.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a document with a paragraph in which text must be oriented vertically, flowing from left to right horizontally on the page.This setting would be specified with the following WordprocessingML:

```
<w:pPr>
  <w:textDirection w:val="lr" />
</w:pPr>
```

The textDirection element specifies via the `lr` value in the val attribute that the text flow must be oriented vertically, with subsequent lines stacked from left to right. *end example*]

Attributes	Description

val (Direction of Text Flow)	Specifies the direction of the text flow for this object.
	
	[*Example*: Consider a document with a section in which text be oriented vertically, flowing from left to right horizontally on the page.This setting requires the following WordprocessingML:
	
	`<w:sectPr>`
	`  …`
	`  <w:textDirection w:val="lr" />`
	`</w:sectPr>`
	
	The textDirection element specifies via the `lr` value in the val attribute that the text flow must be oriented vertically, with subsequent lines stacked from left to right. *end example*]
	
	The possible values for this attribute are defined by the ST_TextDirection simple type (§17.18.93).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextDirection) is located in §A.1. *end note*]

17.3.1.42 top (Paragraph Border Above Identical Paragraphs)

This element specifies the border which shall be displayed above a set of paragraphs which have the same set of paragraph border settings.

To determine if any two adjoining paragraphs shall have an individual top and bottom border or a between border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is identical for all possible paragraphs borders, then the between border is displayed. Otherwise, the final paragraph shall use its bottom border and the following paragraph shall use its top border, respectively. If this border specifies a space attribute, that value determines the space above the text (ignoring any spacing above) which should be left before this border is drawn, specified in points.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied above identical paragraphs.

[*Example*: Consider the following two paragraphs' WordprocessingML definition:

```
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>First paragraph.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:pBdr>
      <w:top w:val="single" w:sz="24" w:space="1" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="B97034"
w:themeColor="accent6" w:themeShade="BF" />
      <w:bottom w:val="single" w:sz="24" w:space="0" w:color="F2DCDB"
w:themeColor="accent2" w:themeTint="33" />
      <w:right w:val="single" w:sz="24" w:space="4" w:color="C3D69B"
w:themeColor="accent3" w:themeTint="99" />
      <w:between w:val="single" w:sz="24" w:space="1" w:color="4F81BD"
w:themeColor="accent1" />
    </w:pBdr>
  </w:pPr>
  <w:r>
    <w:t>Second paragraph.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Since the paragraph border is different between the two paragraphs (the bottom space value goes from 1 to 0), paragraph two uses its top border, which is located one point above the text in that paragraph. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.1.43 topLinePunct (Compress Punctuation at Start of a Line)

This element specifies whether punctuation shall be compressed when it appears as the first character in a line, allowing subsequent characters on the line to be move in accordingly.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then punctuation shall not be compressed in this paragraph, even when it appears at the start of a line.

[*Example*: Consider a paragraph which should allow punctuation at the start of a line to be compressed, in order to prevent it from taking up unnecessary space. This constraint is specified using the following WordprocessingML:

```
<w:pPr>
  <w:topLinePunct w:val="on" />
</w:pPr>
```

The topLinePunct element specifies that this compression must be allowed when displaying this paragraph. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.44 widowControl (Allow First/Last Line to Display on a Separate Page)

This element specifies whether a consumer shall prevent a single line of this paragraph from being displayed on a separate page from the remaining content at display time by moving the line onto the following page.

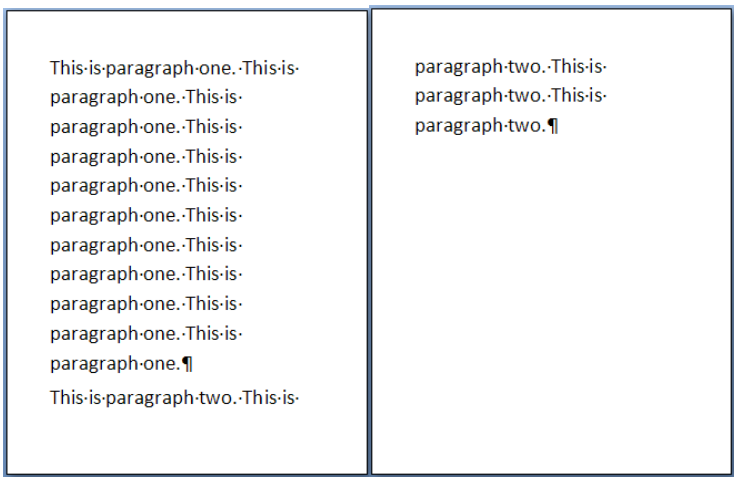
When displaying a paragraph in a page, it is sometimes the case that the first line of that paragraph would display as the last line on one page, and all subsequent lines would display on the following page. This property ensures that a consumer shall move the single line to the following page as well to prevent having one line on its own page. As well, if a single line appears at the top of a page, a consumer shall move the preceding line onto the following page as well, to prevent a single line from being displayed on a separate page.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this paragraph shall prevent a single line from being shown on a separate page whenever it would normally occur.

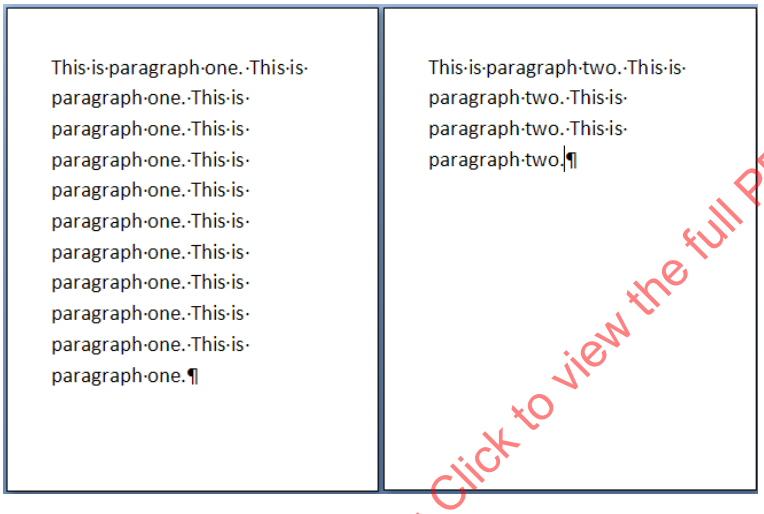
[*Example*: Consider a document with a paragraph which must be shown on four lines at display time. If this paragraph would normally be laid out with its first line at the bottom of one page, and its following lines on the next page, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



This property would ensure that the default behavior for each paragraph prevented this, by moving this line onto the following paragraph as follows:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

However, if this default is overridden by specifying the following WordprocessingML:

```
<w:pPr>
  <w:widowControl w:val="off" />
</w:pPr>
```

The specifying of the widowControl element with value `off` means that the consumer displaying this document must not move the first line onto a separate page if it would be separated from all other lines (the first picture above). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.1.45 wordWrap (Allow Line Breaking At Character Level)

This element specifies whether a consumer shall break text which exceeds the text extents of a line by breaking the word across two lines (breaking on the character level) or by moving the word to the following line (breaking

 ©ISO/IEC 2016 – All rights reserved

on the word level). [*Note*: Applications are discouraged from breaking text when it would change the semantics or appearance of the content. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this paragraph shall break words in space-delimited languages at the word level, not the character level when it is displayed.

[*Example*: Consider a paragraph whose first line ends with the word `world`, where the text extents for that line would normally fall between the letter o and the letter r. If this element is omitted, a producer would normally move the entire word `world` to the following line, since the word does not fit within the first line's text extents. However, if this document should allow words to be broken at the character level, that constraint would be specified as follows:

```
<w:pPr>
  <w:wordWrap w:val="off" />
</w:pPr>
```

The resulting paragraph specifies that wordWrap is turned `off`, therefore the word "world" would be broken into two lines between the exact two characters (o and r) that match the text extents. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2 Run

The next level of the document hierarchy is the *run*, which defines a region of text with a common set of properties. A run is represented by an r element (§17.3.2.25), which allows the producer to specify a single set of formatting properties, applying the same information to all the contents of the run.

Just as a paragraph can have properties, so too can a run. All of the elements inside an r element have their properties controlled by a corresponding optional rPr run properties element (§17.7.9.1; §17.3.2.27), which shall be the first child of the r element. In turn, the rPr element is a container for a set of property elements that are applied to the rest of the children of the r element. [*Note*: The elements inside the rPr container element allow the consumer to control whether the content in the following run content is bold, underlined, or visible, for example. *end note*]

[*Example*: Consider the following run within a WordprocessingML document:

```
<w:r>
  <w:rPr>
    <w:b/>
    <w:i/>
  </w:rPr>
  <w:t>quick</w:t>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The run specifies two formatting properties in its run contents: bold and italic. These properties are therefore applied to all content within this run. *end example*]

17.3.2.1 b (Bold)

This element specifies whether the bold property shall be applied to all non-complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then bold shall not be applied to non-complex script characters.

[*Example*: Consider a run of text which must have the b property explicitly turned off for the non complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:b w:val="false"/>
</w:rPr>
```

This run explicitly declares that the b property is `false` for the non-complex script contents of this run. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.2 bCs (Complex Script Bold)

This element specifies whether the bold property shall be applied to all complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then bold shall not be applied to complex script characters.

[*Example*: Consider a run of text which must have the bCs property (bold) explicitly turned on for the complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:bCs w:val="true"/>
</w:rPr>
```

This run explicitly declares that the bCs property is `true` , so bold is turned on for the complex script contents of this run. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.2.3 bdo (Bidirectional Override)

This element specifies a directional override, which shall be applied as described by the Bidirectional Algorithm (cf. Unicode Technical Report #9). [*Note*: The presence of this markup is functionally equivalent to the presence of a LRO/RLO character at the location of the start element, and a corresponding PDF character at the location of the end element in a string of Unicode text. *end note*]

[*Example*: The right-to-left override can be used to force a product part number made of mixed English, digits, and Hebrew text written from right-to-left, as follows. (Assume lower-case text is English and the upper-case text is Arabic or Hebrew):

```
<w:r>
    <w:t xml:space="preserve">part number: </w:t>
</w:r>
<w:bdo w:val=”rtl”>
    <w:r>
        <w:t>ad-326D-FG</w:t>
    </w:r>
</w:bdo>
```

This generates the following text layout:

```
part number: GF-D623-da.
```

The bidirectional override forces all characters to be displayed right-to-left, regardless of their classification. *end example*]

Attributes	Description

val (Direction of Override)	Specifies the direction of the override being applied.
	
	If this attribute is omitted, the override shall be assumed to be `ltr`.
	
	[*Example*: For example, consider the following bidirectional override:
	
	`<w:bdo w:val="rtl">`
	`…`
	`</w:bdo>`
	
	The val attribute explicitly declares that the override is applied right to left. *end example*]
	
	The possible values for this attribute are defined by the ST_Direction simple type (§17.18.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BdoContentRun) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.3.2.4 bdr (Text Border)

This element specifies information about the border applied to the text in the current run.

The first piece of information specified by the bdr element is that the current shall have a border when displayed. This information is specified simply by the presence of the bdr element in run's properties.

The second piece of information concerns the set of runs which share the current run border. This is determined based on the attributes on the bdr element. If the set of attribute values specifies on two adjacent runs is identical, then those two runs shall be considered to be part of the same run border group and rendered within the same set of borders in the document.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no run border shall be applied to the text in this run.

[*Example*: Consider a document in which the following two runs are located adjacent to one another:

```
<w:r>
  <w:rPr>
    <w:bdr w:val="single" w:sz="36" w:space="0" w:color="B8CCE4"
w:themeColor="accent1" w:themeTint="66" />
  </w:rPr>
  <w:t xml:space="preserve">run one</w:t>
</w:r>
<w:r >
  <w:rPr>
    <w:b />
    <w:bdr w:val="single" w:sz="36" w:space="0" w:color="B8CCE4"
w:themeColor="accent1" w:themeTint="66" />
  </w:rPr>
  <w:t>run two</w:t>
</w:r>
```

These two runs, although each is distinct, are combined when rendering the text border because the bdr elements are identical between the two runs. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.3.2.5 caps (Display All Characters As Capital Letters)

This element specifies that any lowercase characters in this text run shall be formatted for display only as their capital letter character equivalents. This property does not affect any non-alphabetic character in this run, and does not change the Unicode character for lowercase text, only the method in which it is displayed.

This formatting property is a *toggle property* (§17.7.3).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are not formatted as capital letters.

This element shall not be present with the smallCaps (§17.3.2.33) property on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider the words `Hello, world`, which must be displayed in all capital letters in a document. This constraint is specified as follows in the WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:caps w:val="true" />
  </w:rPr>
  <w:t>Hello, world</w:t>
</w:r>
```

This run displays as `HELLO, WORLD`, even though the lowercase characters are used in the run contents due to the use of the caps element. If this property is removed, the original character forms is displayed (they are not lost). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.6 color (Run Content Color)

This element specifies the color which shall be used to display the contents of this run in the document.

This color can be explicitly specified, or set to allow the consumer to automatically choose an appropriate color based on the background color behind the run's content.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are set to allow the consumer to automatically choose an appropriate color based on the background color behind the run's content.

[*Example*: Consider a run of text which should be displayed using the `accent3` theme color from the document's Theme part. This requirement would be specified as follows in the resulting WordprocessingML:

```
<w:rPr>
  <w:color w:themeColor="accent3" />
</w:rPr>
```

The color attribute specifies that the run shall use the `accent3` theme color. *end example*]

Attributes	Description
themeColor (Run	Specifies a theme color which should be applied to the current run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Content Theme Color)	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows for color information to be set centrally in the document.
	
	If the themeColor attribute is specified, then the val attribute is ignored for this run.
	
	[*Example*: Consider a run of text which should be displayed using the `accent3` theme color from the document's Theme part. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:color w:themeColor="accent3" />`
	`</w:rPr>`
	
	The color attribute specifies that the run must use the `accent3` theme color. *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Run Content Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this run's contents. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run. The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border. [*Example*: Consider a shade of 40% applied to a run in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a shade value S (from 0-100), the shade is applied as follows: $$C' = \left(1 - \frac{S}{100}\right) C$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider a document with a run using the `accent6` theme color, whose RGB value (in RRGGBB hex format) is `F79646`. The hex value for the green component is 96 - 150 in decimal. Applying the shade formula with shade of 50%, the output decimal value of the green component is 75, or a hex value of 4B. This transformed value can be seen in the resulting run color WordprocessingML's val attribute: `<w:color w:val="7B4B23" w:themeColor="accent6" w:themeShade="80" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Run Content Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this run's contents. If the themeTint is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run. The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border. [*Example*: Consider a tint of 60% applied to a run in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 = 153 = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a tint value T (from 0-100), the tint is applied as follows: $$C' = \left(1 - \frac{T}{100}\right)(255 - \mathrm{C}) + \mathrm{C}$$ [*Example*: Consider a document with a run using the accent1 theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The hex value for the green component is 50 - 80 in decimal. Applying the tint formula with tint of 60%, the output decimal value of the green component is 150, or a hex value of 96. This transformed value can be seen in the resulting run color's WordprocessingML val attribute:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:color w:val="D99694" w:themeColor="accent1" w:themeTint="99" /> *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Run Content Color)	Specifies the color for this run. This color can either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the run color as appropriate. If the run specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value. [*Example*: Consider a run color with value auto, as follows: <w:rPr> <w:color … w:val="auto" /> </w:rPr> This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the run contents can be distinguished against the page's background color. *end example*] The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.1. *end note*]

17.3.2.7 cs (Use Complex Script Formatting on Run)

This element specifies whether the contents of this run shall be treated as complex script text regardless of their Unicode character values when determining the formatting for this run.

This means that a consumer shall use the complex script formatting applied to the run [*Example*: The bCs value (§17.3.2.2), not the b value (§17.3.2.1). *end example*] when determining the resulting formatting properties.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run contents are set to complex script based on the Unicode character positions of the content.

[*Example*: Consider the following run of English text in a WordprocessingML document:

<w:r>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
  <w:rPr>
    <w:bCs/>
    <w:i/>
    <w:cs/>
  </w:rPr>
  <w:t>some English text</w:t>
</w:r>
```

This run has bold applied to complex script characters, and italics applied to non-complex script characters. However, since the cs property is set, the text in this run must be treated as complex script text when determining the resulting formatting. Therefore, the run has bold formatting, but no italic formatting when displayed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.8 dir (Bidirectional Embedding Level)

This element specifies that the embedding level of its contents shall be increased, as described by the Bidirectional Algorithm (cf. Unicode Technical Report #9). Unlike the rtl element (§17.3.2.30), this element can be nested and causes the embedding level to increase appropriately. Within this element, the semantics for the rtl element shall continue apply, specifying a high-level override of the character classification applied at the appropriate embedding level. [*Note*: The presence of this markup is functionally equivalent to the presence of a LRE/RLE character at the location of the start element, and a corresponding PDF character at the location of the end element in a string of Unicode text. *end note*]

[*Example*: Consider a left-to-right paragraph that has a quote in a right-to-left language, but the quote itself includes left-to-right text. In this case, the text layout within the quote should be read from right-to-left. Adding this element around the quote increases the embedding level and result in the desired layout. (Assume the lower-case text is English and the upper-case text is Arabic or Hebrew):

```
<w:r>
    <w:t>he said: “</w:t>
</w:r>
<w:dir w:val=”rtl”>
    <w:r>
        <w:rPr>
            <w:rtl />
        </w:rPr>
        <w:t>I LEAVE FOR </w:t>
    </w:r>
    <w:r>
        <w:t>united states</w:t>
    </w:r>
    <w:r>
        <w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <w:rtl />
        </w:rPr>
        <w:t xml:space="preserve"> TOMORROW</w:t>
    </w:r>
</w:dir>
<w:r>
    <w:t>”.</w:t>
</w:r>
```

The alternating RTL and LTR elements help in resolving the classification of characters within this run as follows:

```
he said: “I LEAVE FOR united states TOMORROW”.
LLLLLLLLLLRRRRRRRRRRRRLLLLLLLLLLLLLRRRRRRRRRLL
```

But the fact that the quote is enclosed in an RTL embedding; the following embedding levels are generated:

```
he said: “I LEAVE FOR united states TOMORROW”.
00000000001111111111122222222222211111111100
```

This generates the following text layout:

```
he said: “WORROMOT united states ROF EVAEL I”.
```

end example]

Attributes	Description

val (Direction of Embedding)	Specifies the direction of the embedding being applied.
	
	If this attribute is omitted, the embedding shall be assumed to be `ltr`.
	
	[*Example*: For example, consider the following bidirectional embedding:
	
	`<w:dir w:val="rtl">`
	…
	`</w:dir>`
	
	The val attribute explicitly declares that the embedding is applied right to left. *end example*]
	
	The possible values for this attribute are defined by the ST_Direction simple type (§17.18.12).

[*Note*: The W3C XML Schema definition of this element’s content model (CT_DirContentRun) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.2.9 dstrike (Double Strikethrough)

This element specifies that the contents of this run shall be displayed with two horizontal lines through each character displayed on the line.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then double strikethrough shall not be applied to the contents of this run.

This element shall not be present with the strike (§17.3.2.37) property on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the dstrike property explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:dstrike w:val="true"/>
</w:rPr>
```

This run explicitly declares that the dstrike property is `true`, so the contents of this run has two horizontal strikethrough lines. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.10 eastAsianLayout (East Asian Typography Settings)

This element specifies any East Asian typography settings which shall be applied to the contents of the run.

The specific typography settings represented by this element include the *two lines in one* and *horizontal in vertical* text options.

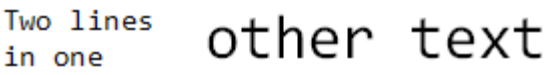
The *two lines in one* setting specifies that the characters in this run should be written out on a single line in the document by creating two sub-lines within the regular line, and laying out this text equally between those sub lines.

[*Example*: Consider a paragraph with the text `two lines in one`, which must be displayed within a single logical line in the document. This constraint would be specified as follows in the WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:eastAsianLayout w:id="1" w:combine="on" />
  </w:rPr>
  <w:t>two lines in one</w:t>
</w:r>
```

The resulting text would be displayed on two sub lines within the other text on this line, like this:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



end example]

The *horizontal in vertical* setting specifies that characters in this run should be rendered with a 90 degree rotation to the left from all other contents of the line when displayed in the document, while keeping the text on the same line as all other text in the paragraph.

[*Example*: Consider a paragraph with the text `this word is vertical`, of which the word `vertical` must be displayed vertically within the document. This constraint would be specified as follows in the WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:eastAsianLayout w:id="2" w:vert="on" />
  </w:rPr>
  <w:t>vertical</w:t>
</w:r>
```

The resulting text would be displayed with a 90 degree rotation from the other text content. *end example*]

Attributes	Description

combine (Two Lines in One)	Specifies whether the contents of the current run should be combined into one line using the two lines in one logic described above in the parent element.
	
	If this attribute is omitted, then this run shall not be displayed on two sub lines.
	
	[*Example*: Consider a paragraph with the text `two lines in one`, which must be displayed within a single logical line in the document. This constraint would be specified as follows in the WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:eastAsianLayout w:id="1" w:combine="on" />`
	`  </w:rPr>`
	`  <w:t>two lines in one</w:t>`
	`</w:r>`
	
	The resulting text would be displayed on two sub lines within the other text on this line. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

combineBrackets (Display Brackets Around Two Lines in	Specifies that the two lines in one text should be enclosed within a pair of brackets when displayed. This attribute's values determine the bracket style to put around combined text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

One)	If this attribute is not specified, then no brackets shall be placed around this content when displayed in the document. If the combine attribute is not specified, then this attribute is ignored.
	
	[*Example*: Consider a paragraph with the text `two lines in one`, which must be displayed within a single logical line in the document and enclosed in curly brackets. This constraint would be specified as follows in the WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:eastAsianLayout w:id="1" w:combine="on" w:combineBrackets="curly"/>`
	`  </w:rPr>`
	`  <w:t>two lines in one</w:t>`
	`</w:r>`
	
	The resulting text would be displayed on two sub-lines within the other text on this line and enclosed within curly brackets when displayed. *end example*]
	
	The possible values for this attribute are defined by the ST_CombineBrackets simple type (§17.18.8).

id (East Asian Typography Run ID)	Specifies a unique ID which shall be used to link multiple runs containing eastAsianLayout element to each other to ensure that their contents are correctly displayed in the document.
	
	This means that multiple runs which are broken apart due to differences in formatting can be identified as belonging to the same grouping in terms of eastAsianLayout properties, although they are separated into multiple runs of text.
	
	[*Example*: Consider the following three runs in a document:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:eastAsianLayout w:id="-1552701694" w:combine="lines" w:combineBrackets="curly" />`
	`  </w:rPr>`
	`  <w:t>two</w:t>`
	`</w:r>`
	`<w:r>`
	`  <w:rPr>`
	`  <w:u w:val="single" w:color="4F81BD" w:themeColor="accent1" />`
	`  <w:eastAsianLayout w:id="-1552701694" w:combine="lines" w:combineBrackets="curly" />`
	`  </w:rPr>`
	`  <w:t>lines in</w:t>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</w:r> <w:r> <w:rPr> <w:eastAsianLayout w:id="-1552701694" w:combine="lines" w:combineBrackets="curly" /> </w:rPr> <w:t>one</w:t> </w:r> Although there are three runs of content, all three regions must be combined into a single two lines in one region based on the identical value used in the id attribute for all three runs. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
vert (Horizontal in Vertical (Rotate Text))	Specifies that characters in this run should be rendered with a 270 degree rotation to the left from all other contents of the line when displayed in the document as described above. If this attribute is omitted, then the contents of this run shall not be rotated with respect to the normal text flow. [*Example*: Consider a paragraph with the text `this word is vertical`, of which the word `vertical` must be displayed vertically within the document. This constraint would be specified as follows in the WordprocessingML: <w:r> <w:rPr> <w:eastAsianLayout w:id="2" w:vert="on" /> </w:rPr> <w:t>vertical</w:t> </w:r> The resulting text would be displayed with a 270 degree rotation from the other text content. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
vertCompress (Compress Rotated Text to Line Height)	Specifies whether the rotated text shall be compressed at display time in order to ensure that it fits into the existing line height without increasing the overall height of the line. If the vert attribute is not specified, then this attribute is ignored. If this attribute is omitted, then text shall not be compressed in order to fit into the existing height of the line when it is rotated. [*Example*: Consider a paragraph with the text `this word is vertical`, of which the word `vertical` must be displayed vertically within the document but must not change

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	the height of the line. This constraint would be specified as follows in the WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:eastAsianLayout w:id="2" w:vert="true"`
	`w:vertCompress="true" />`
	`  </w:rPr>`
	`  <w:t>vertical</w:t>`
	`</w:r>`
	
	The resulting text would be compressed in order to fit the height of the line as defined by all non-compressed characters. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EastAsianLayout) is located in §A.1. *end note*]

17.3.2.11 effect (Animated Text Effect)

This element specifies an animated text effect which should be displayed when rendering the contents of this run. This effect is rendered around the extents of the text in the run in the same location as a run border with zero pixels of padding would be rendered (if such a run border was present).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no text effect shall be applied to the contents of this run.

[*Example*: Consider a run of text which must have an animated text effect consisting of multiple colored flashing lights (see possible attribute values for descriptions of each effect). This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:effect w:val="lights"/>
</w:rPr>
```

This run explicitly declares that the effect property is `lights`, so the contents of this run has an animated lights text effect. *end example*]

Attributes	Description
val (Animated Text Effect Type)	Specifies the type of animated text effect which shall be applied to this text run. [*Example*: Consider a run of text which must have an animated text effect consisting of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	multiple colored flashing lights. This constraint is specified using the following WordprocessingML:
	
	`<w:rPr>`
	`  <w:effect w:val="lights"/>`
	`</w:rPr>`
	
	This run explicitly declares a type of text effect, using the val property, of `lights`, so the contents of this run has the animated lights text effect. *end example*]
	
	The possible values for this attribute are defined by the ST_TextEffect simple type (§17.18.94).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextEffect) is located in §A.1. *end note*]

17.3.2.12 em (Emphasis Mark)

This element specifies the emphasis mark that shall be applied to each non-space character in this run. An *emphasis mark* is an additional character whose display position relative to the character to which it is applied is language- and writing-direction-dependent. The emphasis mark is specified by the contents of the val attribute.

If this element is not present, the default value is to leave the formatting applied at the previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no emphasis mark shall be applied to any character in this run.

[*Example*: Consider a run of text that is to have a dot emphasis mark applied to it. This is specified using the following WordprocessingML:

```
<w:rPr>
  <w:em w:val="dot"/>
</w:rPr>
```

end example]

Attributes	Description
val (Emphasis Mark Type)	Specifies the emphasis mark applied to each non-space character in this run. The possible values for this attribute are defined by the ST_Em simple type (§17.18.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Em) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.2.13 emboss (Embossing)

This element specifies that the contents of this run should be displayed as if embossed, which makes text appear as if it is raised off the page in relief.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then embossing shall not be applied to the contents of this run.

This element shall not be present with either the imprint (§17.3.2.18) or outline (§17.3.2.23) properties on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the emboss property explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:emboss w:val="true"/>
</w:rPr>
```

This run explicitly declares that the emboss property is `true`, so the contents of this run appear embossed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.14 fitText (Manual Run Width)

This element specifies that the contents of this run shall not be automatically displayed based on the width of its contents, rather its contents shall be resized to fit the width specified by the val attribute. This expansion/contraction shall be performed by equally increasing/decreasing the size of each character in this run's contents when displayed.

If this element is omitted, then the contents of this run shall be displayed based on the size of its contents.

[*Example*: Consider a document with a run which must be displayed in exactly one-half inch of space, regardless of its contents. This constraint would be specified using the following WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:fitText w:id="50" w:val="720" />
  </w:rPr>
  <w:t>This text must be displayed in one-half of an inch.</w:t>
</w:r>
```

The resulting run contents must be displayed in exactly `720` twentieths of a point (one half of an inch) when displayed in a document. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

id (Fit Text Run ID)	Specifies a unique ID which shall be used to link multiple contiguous runs containing fitText elements to each other to ensure that their contents are correctly merged into the specified width in the document.
	
	This means that multiple runs which are broken apart due to differences in formatting can be identified as belonging to the same grouping in terms of fitText properties, although they are multiple runs of text in the WordprocessingML.
	
	If the runs are not contiguous, then the id attribute is ignored, and the runs are not linked.
	
	If this attribute is omitted, then this run has no id and shall not be linked with any other run in the parent paragraph.
	
	[*Example*: Consider the following three runs in a document, which should be fit into exactly one inch at display time:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:fitText w:id="99" w:val="1440" />`
	`  </w:rPr>`
	`  <w:t>fit this into</w:t>`
	`</w:r>`
	`<w:r>`
	`  <w:rPr>`
	`  <w:b/>`
	`  <w:fitText w:id="99" w:val="1440" />`
	`  </w:rPr>`
	`  <w:t>one</w:t>`
	`</w:r>`
	`<w:r>`
	`  <w:rPr>`
	`  <w:fitText w:id="99" w:val="1440" />`
	`  </w:rPr>`
	`  <w:t>inch</w:t>`
	`</w:r>`
	
	Although there are three runs of content, all three regions must be combined into a single fit text region (e.g. they all fit into one inch, rather than one inch each) based on the identical value used in the id attribute for all three runs. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

val (Value)	This attribute specifies the exact width of space which this run shall be fit into when displayed in the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a document with a run which must be displayed in exactly one-half inch of space, regardless of its contents. This constraint would be specified using the following WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:fitText w:id="50" w:val="720" />`
	`  </w:rPr>`
	`  <w:t>This text must be displayed in one-half of an inch.</w:t>`
	`</w:r>`
	
	The resulting run contents must be displayed in exactly 720 twentieths of a point (one half of an inch) when displayed in a document. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FitText) is located in §A.1. *end note*]

17.3.2.15 highlight (Text Highlighting)

This element specifies a highlighting color which is applied as a background behind the contents of this run.

If this run has any background shading specified using the shd element (§17.3.2.32), then the background shading shall be superseded by the highlighting color when the contents of this run are displayed.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then text highlighting shall not be applied to the contents of this run.

[*Example*: Consider a run within a paragraph which has run shading applied as well as yellow text highlighting using the highlight element. This formatting is specified using the following WordprocessingML:

```
<w:rPr>
  <w:highlight w:val="yellow" />
  <w:shd w:themeFill="accent2" w:themeFillTint="66" />
</w:rPr>
```

The resulting run would have `yellow` highlighting visible over its contents, as the highlighting supersedes the shading for the contents of the run. *end example*]

Attributes	Description
val (Highlighting	Specifies the color of the text highlighting which shall be applied to the contents of this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Color)	run.
	
	[*Example*: Consider a text run which must be displayed with colored text highlighting. This highlighting would be specified using the following WordprocessingML:
	
	`<w:rPr>`
	`  <w:highlight w:val="red" />`
	`</w:rPr>`
	
	The resulting text highlighting would be `red`, as this is the color specified by the val attribute. *end example*]
	
	The possible values for this attribute are defined by the ST_HighlightColor simple type (§17.18.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Highlight) is located in §A.1. *end note*]

17.3.2.16 i (Italics)

This element specifies whether the italic property should be applied to all non-complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy* .If this element is never applied in the style hierarchy, then italics shall not be applied to non-complex script characters.

[*Example*: Consider a run of text which must have the i property explicitly turned on for the non-complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:i />
</w:rPr>
```

This run explicitly declares that the i property is `true` for the non-complex script contents of this run. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.17 iCs (Complex Script Italics)

This element specifies whether the italic property should be applied to all complex script characters in the contents of this run when displayed in a document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then italics shall not be applied to complex script characters.

[*Example*: Consider a run of text which must have the iCs property explicitly turned on for the complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:iCs w:val="true"/>
</w:rPr>
```

This run explicitly declares that the iCs property is `true`, so italics are turned on for the complex script contents of this run. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.18 imprint (Imprinting)

This element specifies that the contents of this run should be displayed as if imprinted, which makes text appear to be imprinted or pressed into page (also referred to as 'engrave').

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then imprinting shall not be applied to the contents of this run.

This element shall not be present with either the emboss (§17.3.2.13) or outline (§17.3.2.23) properties on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the imprint property explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:imprint w:val="true"/>
</w:rPr>
```

This run explicitly declares that the imprint property is `true`, so the contents of this run appear imprinted. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.19 kern (Font Kerning)

This element specifies whether font kerning shall be applied to the contents of this run. If it is specified, then kerning shall be automatically adjusted when displaying characters in this run as needed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The val attribute specifies the smallest font size which shall have its kerning automatically adjusted if this setting is specified. If the font size in the sz element (§17.3.2.38) is smaller than this value, then no font kerning shall be performed.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then font kerning shall not be applied to the contents of this run.

[*Example*: Consider the following WordprocessingML run which has font kerning properties specified:

```
<w:r>
  <w:rPr>
    <w:kern w:val="28" />
    <w:sz w:val="22" />
  </w:rPr>
</w:r>
```

Even though font kerning is turned on via the kern element, the contents of this run must not be kerned because that settings only applied to font sizes of 14 points (28 half-points) or larger. If the kern element's val attribute was less than or equal to the sz element's val attribute, then kerning would be applied:

```
<w:r>
  <w:rPr>
    <w:kern w:val="22" />
    <w:sz w:val="22" />
  </w:rPr>
</w:r>
```

end example]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: <… w:val="30" /> The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.2.20 lang (Languages for Run Content)

This element specifies the languages which shall be used to check spelling and grammar (if requested) when processing the contents of this run.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the languages for the contents of this run shall be automatically determined based on their contents using any method desired.

[*Example*: Consider a run which contains both Latin and complex script characters in its contents. If those contents should be interpreted as French (Canada) and Hebrew, respectively, that requirement would be specified as follows in the resulting WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:lang w:val="fr-CA" w:bidi="he-IL" />
  </w:rPr>
</w:r>
```

The resulting run specifies that any complex script contents must be spell and grammar checked as if they were Hebrew, and any Latin character contents must be spell and grammar checked as if they were French (Canada). *end example*]

Attributes	Description

bidi (Complex Script Language)	Specifies the language which shall be used when processing the contents of this run which use complex script characters, as determined by the Unicode character values of the run content.
	
	If this attribute is omitted, then the languages for the contents of this run using complex script characters shall be automatically determined based on their contents using any appropriate method.
	
	[*Example*: Consider a run which contains complex script characters in its contents. If those contents should be interpreted as Hebrew, that requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:lang w:bidi="he-IL" />`
	`  </w:rPr>`
	`</w:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The resulting run specifies that any complex script contents must be spell and grammar checked using a Hebrew dictionary and grammar engine, if one is available. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

eastAsia (East Asian Language)	Specifies the language which shall be used when processing the contents of this run which use East Asian characters, as determined by the Unicode character values of the run content.
	
	If this attribute is omitted, then the languages for the contents of this run using East Asian characters shall be automatically determined based on their contents using any appropriate method.
	
	[*Example*: Consider a run which contains East Asian characters in its contents. If those contents should be interpreted as Korean, that requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:lang w:eastAsia="ko-KR" />`
	`  </w:rPr>`
	`</w:r>`
	
	The resulting run specifies that any complex script contents must be spell and grammar checked using a Korean dictionary and grammar engine, if one is available. *end example*]
	
	The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

val (Latin Language)	Specifies the language which shall be used to check spelling and grammar (if requested) when processing the contents of this run which use Latin characters, as determined by the Unicode character values of the run content.
	
	If this attribute is omitted, then the languages for the contents of this run using Latin characters shall be automatically determined based on their contents using any appropriate method.
	
	[*Example*: Consider a run which contains Latin characters in its contents. If those contents should be interpreted as English (Canada), that requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:lang w:val="en-CA" />`
	`  </w:rPr>`
	`</w:r>`
	
	The resulting run specifies that any complex script contents must be spell and grammar checked using a English (Canada) dictionary and grammar engine, if one is available. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	example] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Language) is located in §A.1. *end note*]

17.3.2.21 noProof (Do Not Check Spelling or Grammar)

This element specifies that the contents of this run shall not report any errors when the document is scanned for spelling and grammar. [*Note*: It is entirely at the consumer's/producer's discretion whether this is done by not checking the region for spelling and grammar, or simply by suppressing the results. *end note*]

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then spelling and grammar error shall not be suppressed on the contents of this run.

[*Example*: Consider a run of text which must not ever have spelling or grammar errors reported for the contents of the run, for example, the XML fragments included in ISO/IEC 29500. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:noProof w:val="true"/>
</w:rPr>
```

This run explicitly declares that the noProof property is `true`, so the contents of this run never report spelling or grammar errors. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.22 oMath (Office Open XML Math)

This element specifies that this run contains WordprocessingML which shall be handled as though it was Office Open XML Math.

[*Rationale*: Like other run properties can be applied to the glyph representing the paragraph mark, it is possible to create an Office Open XML Math equation on an empty paragraph as well. Since that paragraph mark must be defined by WordprocessingML, it is not possible to store the paragraph using the Office Open XML Math markup. Instead, this run property is stored on the paragraph mark's run properties to indicate that the paragraph mark is part of an Office Open XML Math equation. For example, the first paragraph below is stored as Office Open XML Math:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

¶

¶

The paragraph must be a p (§17.3.1.22) element, but that would mean the data loss of the Math markup when saving as a WordprocessingML package. In order to prevent that data loss, this property stores the Math property as a run property. *end rationale*]

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then this run shall not be treated as Office Open XML Math.

This property can be applied to any run, but that should only introduce the semantic that the run is math in the user interface, and shall not change the appearance of the text.

[*Example*: Consider a paragraph in WordprocessingML where the paragraph mark glyph (the pilcrow mark - ¶) has been formatted as Math. Since this mark is not an actual run, it cannot be written out in the Office Open XML Math syntax, and must be written out as a property on the actual run as follows:

```
<w:pPr>
  <w:rPr>
    <w:oMath />
  </w:rPr>
</w:pPr>
```

This property is therefore used to roundtrip the math setting on this paragraph mark character. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.23 outline (Display Character Outline)

This element specifies that the contents of this run should be displayed as if they have an outline, by drawing a one pixel wide border around the inside and outside borders of each character glyph in the run.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then outline shall not be applied to the contents of this run.

This element shall not be present with either the emboss (§17.3.2.13) or imprint (§17.3.2.18) properties on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the outline property explicitly turned off for the contents of the run. This constraint is specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:rPr>
  <w:outline w:val="false"/>
</w:rPr>
```

This run explicitly declares that the outline property is `false`, so the contents of this run do not appear as if they have an exterior outline around them. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.24 position (Vertically Raised or Lowered Text)

This element specifies the amount by which text shall be raised or lowered for this run in relation to the default baseline of the surrounding non-positioned text. This allows the text to be repositioned without altering the font size of the contents.

If the val attribute is positive, then the parent run shall be raised above the baseline of the surrounding text by the specified number of half-points. If the val attribute is negative, then the parent run shall be lowered below the baseline of the surrounding text by the specified number of half-points.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall not be raised or lowered relative to the default baseline location for the contents of this run.

[*Example*: Consider a run which must be positioned 12 points above the default baseline location when displaying its contents. This requirement would be specified using the following WordprocessingML:

```
<w:rPr>
  <w:position w:val="24" />
</w:rPr>
```

The resulting run is positioned 24 half-points above the default baseline location because the contents of the val attribute are positive. *end example*]

Attributes	Description

val (Signed Half-Point Measurement)	Specifies a positive or negative measurement in half-points (1/144 of an inch).
	
	The contents of this attribute value are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:rPr>`
	`  <w:position w:val="-12" />`
	`</w:rPr>`
	
	In this case, the value in the val attribute is amount by which the specified run must be raised or lowered compared to the baseline of the surrounding text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	In all cases, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_SignedHpsMeasure simple type (§17.18.80).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedHpsMeasure) is located in §A.1. *end note*]

17.3.2.25 r (Text Run)

This element specifies a run of content in the parent field, hyperlink, custom XML element, structured document tag, smart tag, or paragraph.

The contents of a run in a WordprocessingML document shall consist of any combination of run content.

[*Example*: Consider a basic WordprocessingML paragraph with a pair of runs. This run would be expressed as follows:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Text</w:t>
      </w:r>
      <w:fldSimple w:instr="AUTHOR">
        <w:r>
          <w:t>Author Name</w:t>
        </w:r>
      </w:fldSimple>
    </w:p>
  </w:body>
</w:document>
```

The r element is the container for all of the content in the run, which in this example includes both a run in the paragraph and a run within a simple field. *end example*]

Attributes	Description
rsidDel (Revision Identifier for Run Deletion)	Specifies a unique identifier used to track the editing session when the run was deleted from the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Revision Identifier for Run)	Specifies a unique identifier used to track the editing session when the run was added to the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidRPr (Revision Identifier for Run Properties)	Specifies a unique identifier used to track the editing session when the run properties were last modified in the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_R) is located in §A.1. *end note*]

17.3.2.26 rFonts (Run Fonts)

This element specifies the fonts which shall be used to display the text contents of this run. Within a single run, there can be up to four types of font slot which shall each be allowed to use a unique font:

- ASCII (i.e., the first 128 Unicode code points)
- High ANSI
- Complex Script

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- East Asian

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall be displayed in any default font which supports that set of characters.

[*Example*: Consider a single text run with both Arabic and English text, which can be expressed as follows:

```
<w:r>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:cs="Times New Roman" />
  </w:rPr>
  <w:t>English العربية</w:t>
</w:r>
```

In this run, both “English” and “العربية” should be in ASCII font slot, according to the two-step algorithm below. Therefore, both of them should be in the Courier New font face.

The same content can also be expressed as follows:

```
<w:r>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:cs="Times New Roman" />
    <w:rtl/>
  </w:rPr>
  <w:t>English العربية</w:t>
</w:r>
```

In this run, both “English” and “العربية” should be in Complex Script font slot, according to the two-step algorithm. Therefore, both of them should be in the Times New Roman font face. *end example*]

For each Unicode character in a run, the font slot can be determined using the following two-step methodology:

1. Use the table below to decide the classification of the content, based on its Unicode code point.

Unicode Block	Range	Classification
Basic Latin	U+0000–U+007F	*ASCII font*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Unicode Block	Range	Classification
Latin-1 Supplement	U+00A0–U+00FF	*High ANSI font*, with the following exceptions: • If *the value of the* hint *attribute is* `eastAsia`, the following characters use *East Asian font* (or *eastAsiaTheme* if defined): A1, A4, A7 – A8, AA, AD, AF, B0 – B4, B6 – BA, BC – BF, D7, F7 If *the value of the* hint *attribute is* `eastAsia` and the language component of the language specified in the eastAsia attribute on the lang element is "zh", the following characters use *East Asian font* (or *eastAsiaTheme* if defined): E0 – E1, E8 – EA, EC – ED, F2 – F3, F9 – FA, FC
Latin Extended-A	U+0100–U+017F	*High ANSI font*, with the following exception: If *the value of the* hint *attribute is* `eastAsia`, and the language component of the language specified in the eastAsia attribute on the lang element is "zh", or the character set of the *East Asian font* (or *eastAsiaTheme* if defined) font is Big5 or GB2312, then *East Asian font* is used.
Latin Extended-B	U+0180–U+024F	*High ANSI font*, with the following exception: If *the value of the* hint *attribute is* `eastAsia`, and the language component of the language specified in the eastAsia attribute on the lang element is "zh", or the character set of the *East Asian font* (or *eastAsiaTheme* if defined) font is Big5 or GB2312, then *East Asian* font is used.
IPA Extensions	U+0250–U+02AF	*High ANSI font*, with the following exception: If *the value of the* hint *attribute is* `eastAsia`, and the language component of the language specified in the eastAsia attribute on the lang element is "zh", or the character set of the *East Asian font* (or *eastAsiaTheme* if defined) font is Big5 or GB2312, then *East Asian font* is used.
Spacing Modifier Letters	U+02B0–U+02FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Combining Diacritical Marks	U+0300–U+036F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Greek	U+0370–U+03CF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Cyrillic	U+0400–U+04FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Hebrew	U+0590–U+05FF	*ASCII font*
Arabic	U+0600–U+06FF	*ASCII font*
Syriac	U+0700–U+074F	*ASCII font*
Arabic Supplement	U+0750–U+077F	*ASCII font*
Thaana	U+0780–U+07BF	*ASCII font*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Unicode Block	Range	Classification
Hangul Jamo	U+1100–U+11FF	*East Asian font*
Latin Extended Additional	U+1E00–U+1EFF	*High ANSI font*, with the following exception: If the *value of the* hint *attribute is* `eastAsia` and the language component of the language specified in the eastAsia attribute on the lang element is “zh”, then East Asian is used.
Greek Extended	U+1F00–U+1FFF	*High ANSI font*
General Punctuation	U+2000–U+206F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Superscripts and Subscripts	U+2070–U+209F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Currency Symbols	U+20A0–U+20CF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Combining Diacritical Marks for Symbols	U+20D0–U+20FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Letter-like Symbols	U+2100–U+214F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Number Forms	U+2150–U+218F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Arrows	U+2190–U+21FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Mathematical Operators	U+2200–U+22FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Miscellaneous Technical	U+2300–U+23FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Control Pictures	U+2400–U+243F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Optical Character Recognition	U+2440–U+245F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Enclosed Alphanumerics	U+2460–U+24FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Box Drawing	U+2500–U+257F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Block Elements	U+2580–U+259F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Geometric Shapes	U+25A0–U+25FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
Miscellaneous Symbols	U+2600–U+26FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Unicode Block	Range	Classification
Dingbats	U+2700–U+27BF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
CJK Radicals Supplement	U+2E80–U+2EFF	*East Asian font*
Kangxi Radicals	U+2F00–U+2FDF	*East Asian font*
Ideographic Description Characters	U+2FF0–U+2FFF	*East Asian font*
CJK Symbols and Punctuation	U+3000–U+303F	*East Asian font*
Hiragana	U+3040–U+309F	*East Asian font*
Katakana	U+30A0–U+30FF	*East Asian font*
Bopomofo	U+3100–U+312F	*East Asian font*
Hangul Compatibility Jamo	U+3130–U+318F	*East Asian font*
Kanbun	U+3190–U+319F	*East Asian font*
Enclosed CJK Letters and Months	U+3200–U+32FF	*East Asian font*
CJK Compatibility	U+3300–U+33FF	*East Asian font*
CJK Unified Ideographs Extension A	U+3400–U+4DBF	*East Asian font*
CJK Unified Ideographs	U+4E00–U+9FAF	*East Asian font*
Yi Syllables	U+A000–U+A48F	*East Asian font*
Yi Radicals	U+A490–U+A4CF	*East Asian font*
Hangul Syllables	U+AC00–U+D7AF	*East Asian font*
High Surrogates	U+D800–U+DB7F	*East Asian font*
High Private Use Surrogates	U+DB80–U+DBFF	*East Asian font*
Low Surrogates	U+DC00–U+DFFF	*East Asian font*
Private Use Area	U+E000–U+F8FF	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used, otherwise *High ANSI font* is used.
CJK Compatibility Ideographs	U+F900–U+FAFF	*East Asian font*
Alphabetic Presentation Forms	U+FB00–U+FB4F	If the *value of the* hint *attribute is* `eastAsia` then *East Asian font* is used for characters in the range FB00 – FB1C, otherwise *High ANSI font* is used. For the range FB1D – FB4F, *ASCII font* is used.
Arabic Presentation Forms-A	U+FB50–U+FDFF	*ASCII font*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Unicode Block	Range	Classification
CJK Compatibility Forms	U+FE30–U+FE4F	*East Asian font*
Small Form Variants	U+FE50–U+FE6F	*East Asian font*
Arabic Presentation Forms-B	U+FE70–U+FEFE	*ASCII font*
Halfwidth and Fullwidth Forms	U+FF00–U+FFEF	*East Asian font*

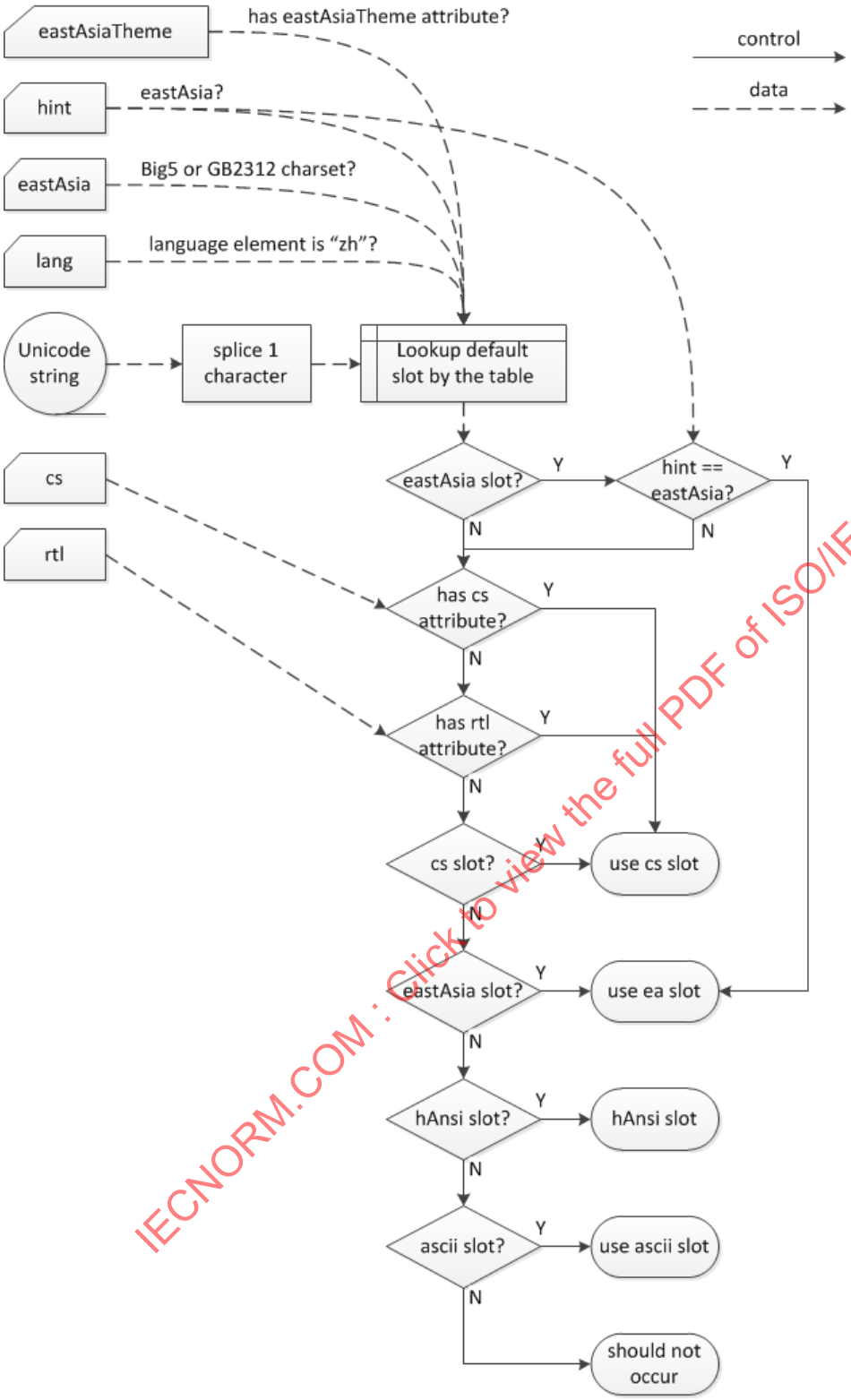
2. If, after the first step, the character falls into East Asian classification and the *value of the* `hint` *attribute is* `eastAsia`, then the character should use East Asian font slot
    a. Otherwise, if there is <w:cs/> or <w:rtl/> in this run, then the character should use Complex Script font slot, regardless of its Unicode code point.
        i. Otherwise, the character is decided using the font slot that is corresponding to the classification in the table above.

Once the font slot for the run has been determined using the above steps, the appropriate formatting elements (either complex script or non-complex script) will affect the content.

[*Note:* This process is also represented in the following diagram:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end note]

[*Example*: Consider text with both English and East Asian character in this WordprocessingML:

```
<w:rPr>
    <w:rFonts  w:hint=”eastAsia”/>
   <w:bCs/>
   <w:rtl/>
</w:rPr>
<w:t> English 中文</w:t>
```

In this example, all the characters in the table are first checked. Because 中文 falls into East Asian classification and plus the *value of the* `hint` *attribute is* `eastAsia` in the run, the characters should use East Asian font slot, and in turn <w:bCs/> (§17.3.2.2) should not be able to make it bold. As to *English*, because it does not fall into the East Asian classification and there is <w:rtl/> in this run, it should use Complex Script font slot, regardless of its Unicode code point. And in turn, <w:bCs/> should be able to make it bold. Therefore, in the document, *English* should be bold, while 中文 should be regular, as illustrated below:

end example]

[*Example*: Consider a single text run with both Arabic and English text, as follows:

Englishالعربية

This content can be expressed in a single WordprocessingML run:

```
<w:r>
  <w:t>Englishالعربية</w:t>
</w:r>
```

Although it is in the same run, the content is in different font faces by specifying a different font for ASCII and CS characters in the run:

```
<w:r>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:cs="Times New Roman" />
  </w:rPr>
  <w:t>English العربية</w:t>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This text run must therefore use the `Courier New` font for all characters in the range U+0000 to U+007F, and must use the `Times New Roman` font for all characters in the Complex Script range. *end example*]

Attributes	Description

ascii (ASCII Font)	Specifies a font which shall be used to format all characters in the Unicode code point range (U+0000–U+007F) within the parent run.
	
	If the asciiTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.
	
	If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports these characters.
	
	[*Example*: Consider a run of text consisting of characters in this range, which must be displayed using the `Courier New` font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:ascii="Courier New" />`
	`</w:rPr>`
	
	The ascii attribute specifies that the run must use the `Courier New` font for all text in thisrange. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

asciiTheme (ASCII Theme Font)	Specifies a theme font which shall be used to format all characters in the Unicode code point range (U+0000–U+007F) within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part,which allows for font information to be set centrally in the document. If the ascii attribute is also specified, then that attribute shall be ignored and this value shall be used instead. If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the ascii attribute. [*Example*: Consider a run of ASCII text which must be displayed using the `majorAscii` theme font. This requirement would be specified as follows in the resulting WordprocessingML: `<w:rPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:rFonts w:asciiTheme="majorAscii" /> </w:rPr> The ascii attribute specifies that the run must use the majorAscii theme font as defined in the document's themes part for all text in this range. *end example*] The possible values for this attribute are defined by the ST_Theme simple type (§17.18.96).
cs (Complex Script Font)	Specifies a font which shall be used to format all characters that are determined to be in the Complex Script font slot within the parent run. If the cstheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead. If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports complex script content. [*Example*: Consider a run of Arabic text which must be displayed using the Arial Unicode MS font. This requirement would be specified as follows in the resulting WordprocessingML: <w:rPr> <w:rFonts w:cs="Arial Unicode MS" /> <w:cs /> </w:rPr> The cs attribute specifies that the run must use the Arial Unicode MS font for all text in a complex script range. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
cstheme (Complex Script Theme Font)	Specifies a theme font which shall be used to format all characters that are determined to be in the Complex Script font slot within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part,which allows for font information to be set centrally in the document. If the cs attribute is also specified, then that attribute shall be ignored and this value shall be used instead. If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the cs attribute. [*Example*: Consider a run of Arabic text that must be displayed using the majorBidi

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	theme font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:cstheme="majorBidi" />`
	`  <w:cs />`
	`</w:rPr>`
	
	The cstheme attribute specifies that the run must use the `majorBidi` theme font as defined in the document's themes part for all text in a complex script range. *end example*]
	
	The possible values for this attribute are defined by the ST_Theme simple type (§17.18.96).

eastAsia (East Asian Font)	Specifies a font which shall be used to format all characters in an East Asian Unicode code point range within the parent run.
	
	If the eastAsiaTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.
	
	If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports East Asian content.
	
	[*Example*: Consider a run of Japanese text which must be displayed using the `MS Mincho` font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:eastAsia="MS Mincho" />`
	`</w:rPr>`
	
	The eastAsia attribute specifies that the run must use the `MS Mincho` font for all text in an East Asian range. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

eastAsiaTheme (East Asian Theme Font)	Specifies a theme font which shall be used to format all characters in an East Asian Unicode code point range within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part,which allows for font information to be set centrally in the document. If the eastAsia attribute is also specified, then that attribute shall be ignored and this value shall be used instead. If this attribute is not present, the default value is to leave the formatting applied at

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the eastAsia attribute.
	
	[*Example*: Consider a run of Japanese text which must be displayed using the `minorEastAsia` theme font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:eastAsiaTheme="minorEastAsia" />`
	`</w:rPr>`
	
	The eastAsiaTheme attribute specifies that the run must use the `minorEastAsia` theme font as defined in the document's themes part for all text in an East Asian range. *end example*]
	
	The possible values for this attribute are defined by the ST_Theme simple type (§17.18.96).

hAnsi (High ANSI Font)	Specifies a font which shall be used to format all characters in a Unicode code point range within the parent run which does not fall into one of the three categories defined above, which is called the *high ANSI* range in WordprocessingML.
	
	If the hAnsiTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.
	
	If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports high ANSI content.
	
	[*Example*: Consider a run of text which falls into a high ANSI range, and must be displayed using the `Bauhaus 93` font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:hAnsi="Bauhaus 93" />`
	`</w:rPr>`
	
	The hAnsi attribute specifies that the run must use the `Bauhaus 93` font for all text in a high ANSI range. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

hAnsiTheme (High ANSI Theme Font)	Specifies a theme font which shall be used to format all characters in a Unicode code point range within the parent run which does not fall into one of the three categories defined above, which is called the *high ANSI* range in WordprocessingML. This theme font is a reference to one of the predefined theme fonts, located in the document's

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	Theme part,which allows for font information to be set centrally in the document.
	
	If the hAnsi attribute is also specified, then that attribute shall be ignored and this value shall be used instead.
	
	If this attribute is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the hAnsi attribute.
	
	[*Example*: Consider a run of text which falls into a high ANSI range, and must be displayed using the `minorHAnsi` theme font. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:rFonts w:hAnsiTheme="minorHAnsi" />`
	`</w:rPr>`
	
	The hAnsiTheme attribute specifies that the run must use the `minorHAnsi` theme font as defined in the document's themes part for all text in a high ANSI range. *end example*]
	
	The possible values for this attribute are defined by the ST_Theme simple type (§17.18.96).

hint (Font Content Type)	Specifies the font type which shall be used to format any ambiguous characters in the current run.
	
	There are certain characters which are not explicitly stored in the document, and can be mapped into multiple categories of the four mentioned above. This attribute shall be used to arbitrate that conflict, and determine how ambiguities in this run shall be handled. [*Note*: This is primarily used to handle the formatting on the paragraph mark glyph, and other characters that are not stored as text in the WordprocessingML document. *end note*]
	
	If this attribute is omitted, then this ambiguity can be resolved by any means available.
	
	[*Example*: Consider two runs, both of which contains an ellipsis in the text but the hint is different. The first run is specified as follows in the WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:rFonts/>`
	`  </w:rPr>`
	`  <w:t>Ellipsis…</w:t>`
	`</w:r>`
	
	This piece of text would be displayed as below in a document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	Ellipsis… The second run is specified as follows in the WordprocessingML: <w:r> <w:rPr> <w:rFonts w:hint="eastAsia" /> </w:rPr> <w:t>省略…</w:t> </w:r> This piece of text would be displayed as below in a document: 省略… Although the "…" in both runs has the same Unicode code point, the first run uses ASCII font slot, while the second run uses the East Asian font slot, as determined by the hint attribute. Therefore, these two ellipses look different in the document. *end example*] [*Example*: Consider the run representing the paragraph mark glyph, which is not stored as a physical character. Since this could therefore be formatted with any of the fonts specified for the run, this ambiguity is resolved using the following WordprocessingML: <w:pPr> <w:rPr> <w:rFonts w:hint="eastAsia" /> </w:rPr> </w:pPr> The hint attribute specifies that the run must use the eastAsia font (theme or not, whichever is in use for East Asian text) as defined for this range. *end example*] The possible values for this attribute are defined by the ST_Hint simple type (§17.18.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fonts) is located in §A.1. *end note*]

17.3.2.27 rPr (Previous Run Properties)

This element specifies a set of run properties which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

[*Example*: Consider a run which has a set of run formatting properties that were added with revision tracking turned on. This set of revised properties is specified in the run properties as follows:

<w:r>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:rPr>
      <w:b />
      <w:imprint />
      <w:lang w:val="en-CA" />
      <w:rPrChange … w:author="user1">
        <w:rPr>
          <w:i />
          <w:dstrike w:val="false" />
        </w:rPr>
      </w:rPrChange>
    </w:rPr>
  </w:r>
```

The rPr element under rPrChange specifies the properties which were applied to the current run before revision tracking was turned on - in this case, italics using the i element (§17.3.2.16), and that any double strikethrough which was applied based on the style hierarchy must be turned off using the dstrike element (§17.3.2.9). *end example*]

The W3C XML Schema definition of this element's content model (CT_RPrOriginal) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

17.3.2.28 rPr (Run Properties)

This element specifies a set of run properties which shall be applied to the contents of the parent run after all style formatting has been applied to the text. These properties are defined as *direct formatting*, since they are directly applied to the run and supersede any formatting from styles.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles
- Numbering styles
- Paragraph styles
- Character styles
- Direct formatting (this element)

[*Example*: Consider a run which should have a set of run formatting properties. This set of properties is specified in the run properties as follows:

```
<w:r>
  <w:rPr>
    <w:b />
    <w:imprint />
    <w:lang w:val="en-CA" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </w:rPr>
  </w:r>
```

The rPr element specifies the properties which are applied to the current run - in this case, bold formatting on the run contents using the b element (§17.3.2.1), an imprinted (engraved) text effect using the imprint element (§17.3.2.18), and that this text should be interpreted as English (Canada) when spell or grammar checking the run text using the lang element (§17.3.2.20). *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

17.3.2.29 rStyle (Referenced Character Style)

This element specifies the style ID of the character style which shall be used to format the contents of this paragraph.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles
- Numbering styles
- Paragraph styles
- Character styles (this element)
- Direct Formatting

This means that all properties specified in the style element (§17.7.4.17) with a styleId which corresponds to the value in this element's val attribute are applied to the run at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no character style shall be applied to the current paragraph. As well, this property is ignored if the run properties are part of a character style.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:rPr>
  <w:pStyle w:val="TestCharacterStyle" />
  <w:b />
  <w:i />
</w:rPr>
```

This run specifies that it inherits all of the run properties specified by the paragraph style with a styleId of `TestCharacterStyle`, which then have any bold or italics settings overridden and set to be applied to the run. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.3.2.30 rtl (Right To Left Text)

This element specifies whether the contents of this run shall have right-to-left characteristics. Specifically, the following behaviors are applied when this element's val attribute is `true` (or an equivalent):

- Formatting – When the contents of this run are displayed, all characters shall be treated as complex script characters. This means that the values of the bCs element (§17.3.2.2) and the iCs element (§17.3.2.17) shall be use to determine bold and italic formatting, that the cs/cstheme attributes on the rFonts element (§17.3.2.26) shall be used to determine the font face, and the szCs element (§17.3.2.39) shall be used to determine the font size.
- Character Directionality Override – When the contents of this run are displayed, this property acts as a right-to-left override for characters which are classified as follows (using the Unicode Character Database):
  - Weak types except European Number, European Number Terminator, Common Number Separator, Arabic Number and (for Hebrew text) European Number Separator when constituting part of a number

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Neutral types

- [*Rationale*: This override allows applications to store and utilize higher-level information beyond that implicitly derived from the Unicode Bidirectional algorithm. For example, if the string “first second” appears in a right-to-left paragraph inside a document, the Unicode algorithm would always result in “first second” at display time (since the neutral character is surrounded by strongly classified characters). However, if the whitespace was entered using a right-to-left input method (e.g. a Hebrew keyboard), then that character could be classified as RTL using this property, allowing the display of “second first” in a right-to-left paragraph, since the user explicitly asked for the space in a right-to-left context. *end rationale*]

  This element provides information used to resolve the (Unicode) classifications of individual characters as either L, R, AN or EN. Once this is determined, the line should be displayed subject to the recommendation of the Unicode BiDi algorithm in reordering resolved levels.

This property shall not be used with strongly left-to-right text. Any behavior under that condition is unspecified. . This property, when off, shall not be used with strong right-to-left text. Any behavior under that condition is unspecified.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then right to left characteristics shall not be applied to the contents of this run.

[*Example*: Consider the following WordprocessingML visual content: “`first second,` أول ثاني”. This content might appear as follows within its parent paragraph:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">first second, </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rtl/>
    </w:rPr>
    <w:t>اول ثاني
</w:t>
  </w:r>
</w:p>
```

The presence of the rtl element on the second run specifies that:

- The formatting on that run is specified using the complex-script property variants.
- The whitespace character is treated as right-to-left.

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.31 shadow (Shadow)

This element specifies that the contents of this run shall be displayed as if each character has a shadow. For left-to-right text, the shadow is beneath the text and to its right; for right-to-left text, the shadow is beneath the text and to its left.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then shadowing shall not be applied to the contents of this run.

This element shall not be present with either the emboss (§17.3.2.13) or imprint (§17.3.2.18) properties on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the shadow property explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:shadow w:val="true"/>
</w:rPr>
```

This run explicitly declares that the shadow property is true, so the contents of this run appear with a shadow. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.32 shd (Run Shading)

Like paragraph shading, this element specifies the shading applied to the contents of the run.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then run shading shall not be applied to the contents of this run.

[*Example*: Consider a run which must have a background consisting of a theme color `accent6` with a theme color `text2` overlaid using a 20% fill pattern. This requirement is specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:pPr>
  <w:shd w:val="pct20" w:themeColor="text2" w:themeFill="accent6" />
</w:pPr>
```

The resulting run uses the background color `accent6` under the foreground pattern color `text2` as specified by the `pct20` pattern mask. *end example*]

This element's content model is defined by the common shading properties definition in §17.3.5.

17.3.2.33 smallCaps (Small Caps)

This element specifies that all small letter characters in this text run shall be formatted for display only as their capital letter character equivalents in a font size two points smaller than the actual font size specified for this text. This property does not affect any non-alphabetic character in this run, and does not change the Unicode character for lowercase text, only the method in which it is displayed. If this font cannot be made two point smaller than the current size, then it shall be displayed as the smallest possible font size in capital letters.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are not formatted as capital letters.

This element shall not be present with the caps (§17.3.2.5) property on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider the words `Hello, world`, which must be displayed in small capital letters in a document. This constraint is specified as follows in the WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:smallCaps w:val="true" />
    <w:sz w:val="24" />
  </w:rPr>
  <w:t>Hello, world</w:t>
</w:r>
```

This run displays using a 12 point capital letter for the capital letter H and W, and a 10 point capital letter for the lowercase letters in the run, even though the lowercase characters are used in actual run contents. If this property is removed, the original character forms is displayed (they are not lost). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.34 snapToGrid (Use Document Grid Settings For Inter-Character Spacing)

This element specifies whether the current run should use the document grid characters per line settings defined in the docGrid element (§17.6.5) when laying out the contents in this run. This setting determines

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

whether the additional character pitch specified in the document grid shall be added to each character in this run as specified by the document grid.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall use the document grid setting to lay out text when a document grid is defined for the parent section.

[*Example*: Consider two runs in a section with a document grid set to allow 20 characters per line. This document grid would effectively specifies that an additional character pitch must be added to each line in order to ensure that the resulting line contains only 20 East Asian characters.

If this property is set on the first run, but turned off on the second run, as follows:

```
<w:r>
  <w:t>Run One</w:t>
</w:r>
<w:r>
  <w:rPr>
    <w:snapToGrid w:val="off" />
  </w:rPr>
  <w:t>Run Two</w:t>
</w:r>
```

The resulting document must have the required additional character pitch added to each character in run one, but zero additional character pitch added to each character in run two, since the snapToGrid property is turned off. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.35 spacing (Character Spacing Adjustment)

This element specifies the amount of character pitch which shall be added or removed after each character in this run before the following character is rendered in the document. This property has an effect equivalent to the additional character pitched added by a document grid applied to the contents of a run.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall not have any additional character pitch applied to any character in its contents.

[*Example*: Consider a run of text which must have ten points of additional character spacing explicitly added to each character within the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:spacing w:val="200"/>
</w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This run explicitly declares that the spacing value is 200, so the contents of this run appear as if they have 10 additional points of spacing added between them. *end example*]

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of -720 whose type is ST_SignedTwipsMeasure. This attribute value specifies a value of negative one-half of an inch or -36 points (-720 twentieths of a point = -36 points = -0.5 inches). *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedTwipsMeasure) is located in §A.1. *end note*]

17.3.2.36 specVanish (Paragraph Mark Is Always Hidden)

This element specifies that the given run shall always behave as if it is hidden, even when hidden text is being displayed in the current document.

This property shall only be used to specify that a paragraph mark shall never be used to break the end of a paragraph for display, even if it is being shown on the document, as would be the case if a regularly hidden paragraph was not being displayed in the document. [*Note*: This property was typically used to ensure that a paragraph style can be applied to a part of a paragraph, and still appear as in the Table of Contents (which in previous word processors would ignore the use of the style if it were being used as a character style. *end note*] If this element is applied to any other run, it can be ignored.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run properties for the paragraph mark shall not always be treated as if hidden.

[*Example*: Consider a paragraph mark which never be used to break the end of the paragraph in the document. This constraint is specified using the following WordprocessingML:

```
<w:pPr>
  <w:rPr>
    <w:specVanish />
  </w:rPr>
</w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The presence of the specVanish element means that this paragraph mark must always be treated as hidden (must never be used to end the paragraph for display), but can be used to mark the end of use of a paragraph style. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.37 strike (Single Strikethrough)

This element specifies that the contents of this run shall be displayed with a single horizontal line through the center of the line.

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then strikethrough shall not be applied to the contents of this run.

This element shall not be present with the dstrike (§17.3.2.9) property on the same run, since they are mutually exclusive in terms of appearance.

[*Example*: Consider a run of text which must have the strike property explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:strike w:val="true"/>
</w:rPr>
```

This run explicitly declares that the strike property is `true`, so the contents of this run has a single horizontal strikethrough line. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.38 sz (Non-Complex Script Font Size)

This element specifies the font size which shall be applied to all non complex script characters in the contents of this run when displayed. The font sizes specified by this element's val attribute are expressed as half-point values.

If this element is not present, the default is to leave the font size at the value applied at the previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then any appropriate font size can be used for non complex script characters.

[*Example*: Consider a run of text which must have an explicit font size of 13.5 points for the non complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:sz w:val="27"/>
</w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This run explicitly declares that the sz property is 27 half-point for the non-complex script contents of this run, so the text is displayed in 13.5 point font size. *end example*]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.2.39 szCs (Complex Script Font Size)

This element specifies the font size which shall be applied to all complex script characters in the contents of this run when displayed. The font sizes specified by this element's val attribute are expressed as half-point values.

If this element is not present, the default is to leave the font size at the value applied at the previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then any appropriate font size can be used for complex script characters.

[*Example*: Consider a run of text which must have an explicit font size of 10 points for the complex script contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:szCs w:val="20"/>
</w:rPr>
```

This run explicitly declares that the sz property is 20 half-point for the non-complex script contents of this run, so the text is displayed in 10 point font size. *end example*]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is `30`, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.2.40 u (Underline)

This element specifies that the contents of this run should be displayed along with an underline appearing directly below the character height (less all spacing above and below the characters on the line).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then an underline shall not be applied to the contents of this run.

[*Example*: Consider a run of text which must have a double underline explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:u w:val="double"/>
</w:rPr>
```

This run explicitly declares an underline using the u property. The val of that underline is `double`, so the style of the underline on this run must be a double line. *end example*]

Attributes	Description
color (Underline Color)	Specifies the color for the underlining on this run. This color can either be presented as a hex value (in RRGGBB format), or `auto` to allow a consumer to automatically determine the underline color as appropriate. If the underline specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a run color with value `auto`, as follows:
	
	`<w:rPr>`
	`  <w:u … w:color="auto" />`
	`</w:rPr>`
	
	This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the underline can be distinguished against the page's background color. *end example*]
	
	The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).

themeColor (Underline Theme Color)	Specifies a theme color which should be applied to the current underline.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows for color information to be set centrally in the document.
	
	If the themeColor attribute is specified, then the color attribute is ignored for this underline.
	
	[*Example*: Consider an underlined run of text whose underline should be displayed using the `accent3` theme color from the document's Theme part. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:u … w:themeColor="accent3" />`
	`</w:rPr>`
	
	The themeColor attribute specifies that the underline must use the `accent3` theme color. *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Underline Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this underline. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this underline. The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border. [*Example*: Consider a shade of 40% applied to a underline in a document. This shade is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	calculated as follows: $$S_{xml} = 0.4 * 255 \\ = 102 \\ = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Shade}_{\mathrm{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get `943634`. This transformed value can be seen in the resulting underline's color attribute: `<w:u w:color="943634" w:themeColor="accent2" w:themeShade="BF" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Underline Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this underline's contents. If the themeTint is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border. [*Example*: Consider a tint of 60% applied to an underline in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 \\ = 153 \\ = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Tint}_{\mathrm{pct}} + \left(1 - \mathrm{Tint}_{\mathrm{pct}}\right)$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `4F81BD`. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6) \\ = 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get `95B3D7`. This transformed value can be seen in the resulting underline formatting's WordprocessingML color attribute: `<w:u … w:color="95B3D7" w:themeColor="accent2" w:themeTint="99" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Underline Style)	Specifies the pattern which shall be used to create the underline applied beneath the text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	in this run.
	
	Each of these possible patterns are shown in the simple type referenced below.
	
	If this attribute is omitted, its value is determined by the setting previously set at any level of the style hierarchy. If this setting is never specified in the style hierarchy, then its value shall be "auto".
	
	[*Example*: Consider a run of text which must have a double underline explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:
	
	`<w:rPr>`
	`  <w:u w:val="double"/>`
	`</w:rPr>`
	
	The val of the underline on this run is `double`, so the style of the underline on this run must be a double line. *end example*]
	
	The possible values for this attribute are defined by the ST_Underline simple type (§17.18.99).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Underline) is located in §A.1. *end note*]

17.3.2.41 vanish (Hidden Text)

This element specifies whether the contents of this run shall be hidden from display at display time in a document. [*Note*: The setting should affect the normal display of text, but an application can have settings to force hidden text to be displayed. *end note*]

This formatting property is a *toggle property* (§17.7.3).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy* .If this element is never applied in the style hierarchy, then this text shall not be hidden when displayed in a document.

[*Example*: Consider a run of text which must have the hidden text property turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:vanish />
</w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This run declares that the vanish property is set for the contents of this run, so the contents of this run is hidden when the document contents are displayed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.2.42 vertAlign (Subscript/Superscript Text)

This element specifies the alignment which shall be applied to the contents of this run in relation to the default appearance of the run's text. This allows the text to be repositioned as subscript or superscript without altering the font size of the run properties.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall not be subscript or superscript relative to the default baseline location for the contents of this run.

[*Example*: Consider a run which must be positioning as superscript when displaying its contents. This requirement would be specified using the following WordprocessingML:

```
<w:rPr>
  <w:vertAlign w:val="superscript" />
</w:rPr>
```

The resulting run is positioned as `superscript`, therefore it is rendered in a smaller size above the default baseline location for the contents of the run. *end example*]

Attributes	Description

val (Subscript/Superscript Value)	Specifies the type of vertical alignment applied to the contents of the current run.
	
	[*Example*: Consider a run which must be positioning as superscript when displaying its contents. This requirement would be specified using the following WordprocessingML:
	
	`<w:rPr>`
	`  <w:vertAlign w:val="superscript" />`
	`</w:rPr>`
	
	The value of the val attribute is `superscript`, therefore the run's contents are rendered in a smaller size above the default baseline location for the contents of the run. *end example*]
	
	The possible values for this attribute are defined by the ST_VerticalAlignRun simple type (§22.9.2.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VerticalAlignRun) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.3.2.43 w (Expanded/Compressed Text)

This element specifies the amount by which each character shall be expanded or when the character is rendered in the document. This property has an of stretching or compressing each character in the run, as opposed to the spacing element (§17.3.2.35) which expands/compresses the text by adding additional character pitch but not changing the width of the actual characters displayed on the line.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall be displayed at 100% of its normal width.

[*Example*: Consider a run of text which must be expanded to 200% of its normal width when displaying each character within the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:w w:val="200%"/>
</w:rPr>
```

This run explicitly declares that the w value is 200%, so the contents of this run appear at 200% of their normal character width by stretching the width of each character. *end example*]

Attributes	Description

val (Text Expansion/Compression Value)	Specifies that the percentage by which the contents of this run shall be expanded or compressed with respect to its normal (100%) character width.
	
	If this attribute is omitted, then the contents of this run shall be displayed at 100% of its normal size.
	
	[*Example*: Consider a run of text which must be compressed to 200% when displaying each character within the contents of the run. This constraint is specified using the following WordprocessingML:
	
	`<w:rPr>`
	`  <w:w w:val="50%"/>`
	`</w:rPr>`
	
	This run explicitly declares that the w value is 50%, so the contents of this run appear at 50% of their normal character width by compressing the width of each character. *end example*]
	
	The possible values for this attribute are defined by the ST_TextScale simple type (§17.18.95).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextScale) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.3.2.44 webHidden (Web Hidden Text)

This element specifies whether the contents of this run shall be hidden from display at display time in a document when the document is being displayed in a web page view (§17.18.102). [*Note*: The setting should affect the normal display of text in a web page view, but an application can have settings to force hidden text to be displayed. *end note*]

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy* .If this element is never applied in the style hierarchy, this text shall not be hidden when displayed in a document in a web page view.

[*Example*: Consider a run of text which must have the hidden text property turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:webHidden />
</w:rPr>
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.3 Run Content

The final level of the document hierarchy is *run content*, which is defined as the set of elements which can be contained as the contents of a particular run in a document.

[*Note*: Types of run content in WordprocessingML include:

- Text
- Field Codes
- DrawingML objects
- Fields

end note]

[*Example*: Consider the following run within a WordprocessingML document:

```
<w:r>
  <w:rPr>
    <w:b/>
    <w:i/>
  </w:rPr>
  <w:t>quick</w:t>
</w:r>
```

The run content consists of a single string of run text inside the t element, which reads `quick`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.3.3.1 br (Break)

This element specifies that a break shall be placed at the current location in the run content. A *break* is a special character which is used to override the normal line breaking that would be performed based on the normal layout of the document's contents. [*Example*: Normal breaking for English would occur only after a breaking space or optional hyphen character. *end example*]

The behavior of this break character (the location where text shall be restarted after this break) shall be determined by its type and clear attribute values, described below.

[*Example*: Consider the following sentence in a WordprocessingML document:

```
This is a simple sentence.
```

Normally, just as shown above, this sentence would be displayed on a single line as it is not long enough to require line breaking (given the width of the current page). However, if a text wrapping break character (a typical line break) were inserted after the word `is`, as follows:

```
<w:r>
  <w:t>This is</w:t>
  <w:br/>
  <w:t xml:space="preserve"> a simple sentence.</w:t>
</w:r>
```

This would imply that this break must be treated as a simple line break, and break the line after that word:

```
This is
 a simple sentence.
```

The break character forced the following text to be restarted on the next available line in the document. *end example*]

Attributes	Description
clear (Restart Location For Text Wrapping Break)	Specifies the location which shall be used as the next available line when the break's type attribute has a value of `textWrapping`. This property only affects the restart location when the current run is being displayed on a line which does not span the full text extents due to the presence of a floating object (see possible values for details). If this break is not of style `textWrapping`, then this attribute shall be ignored. If this attribute is omitted, then its value shall be assumed to be none if needed. [*Example*: Consider a text wrapping break character which should force the restart location to the next line which spans the full width of the text extents of the page (there are no floating objects which interrupt the line). This line break is of style `textWrapping`, since it must only advance to the next line, but the clear value must specify that this restart location must ignore all lines which are not

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	of the full line width by specifying a value of `all`, as follows: `<w:br w:type="textWrapping" w:clear="all" />` This break must therefore not use the next available line, but rather the next available line ignoring all lines which do not span the full text width. *end example*] The possible values for this attribute are defined by the ST_BrClear simple type (§17.18.3).
type (Break Type)	Specifies the break type of the current break. The break type determines the next location where text shall be placed after this manual break is applied to the text contents (see possible values for details). If this attribute is omitted, then it shall be assumed to be of style `textWrapping`. [*Example*: Consider a manual break which must advance the text to the next text column in the document, rather than just the next available line. This break would therefore be specified as follows: `<w:br w:type="column"/>` The type attribute specifies a value of `column`, which means that the break must force the next character in the document to be restarted on the next line in a new text column in the document. *end example*] The possible values for this attribute are defined by the ST_BrType simple type (§17.18.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Br) is located in §A.1. *end note*]

17.3.3.2 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ISO/IEC 29500. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[*Example*: Consider a WordprocessingML document which includes the following MathML markup in a part named math1.xml:

```
<mml:math xmlns:mml="http://www.w3.org/1998/Math/MathML">
  <mml:mi>r</mml:mi>
  <mml:mo>=</mml:mo>
  <mml:msup>
    <mml:mrow>
      <mml:mfenced separators="|">
        <mml:mrow>
          <mml:mi>V</mml:mi>
          <mml:mfrac>
            <mml:mrow>
              <mml:mn>3</mml:mn>
            </mml:mrow>
            <mml:mrow>
              <mml:mn>4</mml:mn>
              <mml:mi>π</mml:mi>
            </mml:mrow>
          </mml:mfrac>
        </mml:mrow>
      </mml:mfenced>
    </mml:mrow>
    <mml:mrow>
      <mml:mfrac>
        <mml:mrow>
          <mml:mn>1</mml:mn>
        </mml:mrow>
        <mml:mrow>
          <mml:mn>3</mml:mn>
        </mml:mrow>
      </mml:mfrac>
    </mml:mrow>
  </mml:msup>
</mml:math>
```

The Main Document Part would reference this content as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:body>
  …
  <w:p>
    <w:r>
      <w:contentPart r:id="rId8" />
    </w:r>
  </w:p>
  …
</w:body>
```

The contentPart element specifies that the content targeted by the relationship with an ID of rId8 is part of the WordprocessingML document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="rId8" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/customXml"
Target="math1.xml" />
  …
</Relationships>
```

The corresponding relationship part item shows that the file to be imported is located next to the main document and is named `math1.xml`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… r:id="rId10" /> The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.3.3.3 control (Embedded Control)

This element specifies that the parent embedded object is a representation of an embedded control. This element shall be used to associate the appropriate embedded control settings and properties when the document is displayed.

If the embedded control is not present, cannot be loaded due to application settings, or is not supported, then a suitable placeholder image shall be used to provide a representation of the presence of an embedded control at the appropriate location in the document.

[*Example*: Consider a run which consists of an embedded control. That run would be specified using the following WordprocessingML:

```
<w:r>
  <w:object>
    …
    <w:control r:id="rId99" w:shapeid="10" … />
  </w:object>
</w:r>
```

The control element indicates that the parent embedded object is an embedded control, whose settings and properties are stored on this element and the (optional) target of the relationship specified using the id attribute. *end example*]

Attributes	Description
id (Embedded Control Properties Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationshi	Specifies the relationship ID for the relationship which contains the properties for this embedded control. This property bag is contained in a separate part within the Office Open XML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/control or the document shall be conisdered non-conformant. If this attribute is omitted, then the embedded control shall be given no property bag

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
ps	when instantiated. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="_x0000_s1027" />` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the property data for this embedded control. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Unique Name for Embedded Control)	Specifies a unique name for this embedded control. This name shall be unique across all controls in this document. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="_x0000_s1027" />` The name attribute specifies that the unique name for this control must be `CheckBox1`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
shapeid (Shape Reference)	Specifies the shape ID for a shape which shall be used to define the presentation and location of this embedded control within the document if the control is floating using the DrawingML syntax. [*Note*: This positioning data is sufficient to display the control in any case where: • The embedded control is not on the current machine • Embedded controls are disabled • Embedded controls of this control type are not supported *end note*] This shape ID reference is resolved by looking for a DrawingML object whose id attribute matches the value specified within this attribute. If no such shape exists, then the control shall be rendered inline in the document content at the current run content location. If this attribute is omitted, then this embedded control shall be displayed inline in the current location in the parent run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:name="CheckBox1" w:shapeid="10" />` The shapeid attribute specifies that the DrawingML object with an id attribute value of 10 must contain the positioning data for this embedded control. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Control) is located in §A.1. *end note*]

17.3.3.4 cr (Carriage Return)

This element specifies that a carriage return shall be placed at the current location in the run content. A *carriage return* is the equivalent of Unicode character 000D, and is used to end the current line of text in WordprocessingML.

The behavior of a carriage return in run content shall be identical to a break character with null type and clear attributes, which shall end the current line and find the next available line on which to continue.

[*Example*: Consider the following sentence in a WordprocessingML document:

```
This is another simple sentence.
```

Normally, just as shown above, this sentence would be displayed on a single line as it is not long enough to require line breaking (given the width of the current page). However, if a carriage return were inserted after the word another, as follows:

```
<w:r>
  <w:t>This is another</w:t>
  <w:cr/>
  <w:t xml:space="preserve"> simple sentence.</w:t>
</w:r>
```

This would imply that this carriage return character must force a line break, and break the line after that word:

```
This is another
 simple sentence.
```

The carriage return character forced the following text to be restarted on the next available line in the document. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.5 dayLong (Date Block - Long Day Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs) . [*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

The long day format date block shall use a date picture of DDDD, retrieving the long day format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is a long date: </w:t>
  <w:dayLong />
</w:r>
```

This run specifies that a long day format date block must be placed after the text string literal `This is a long date:` in the document. Assuming that the host application's primary editing language is French (Canada) and today's date is 2006-04-12, this run would be displayed as follows:

```
This is a long date: mercredi
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.6 dayShort (Date Block - Short Day Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs) . [*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

The short day format date block shall use a date picture of DD, retrieving the short day format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:t xml:space="preserve">This is a short date: </w:t>
  <w:dayShort />
</w:r>
```

This run specifies that a short day format date block must be placed after the text string literal `This is a short date:` in the document. Assuming that the host application's primary editing language is English (Canada) and today's date is 2006-04-12, this run would be displayed as follows:

```
This is a short date: 12
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.7 delText (Deleted Text)

This element specifies that this run contains literal text which shall be displayed in the document. The `delText` element shall be used for all text runs which are part of a region of text that is contained in a deleted region using the del element (§17.13.5.14).

[*Example*: Consider a paragraph of WordprocessingML content which reads `This is deleted text`, where the words `deleted text` are part of a deleted region of the document. This paragraph would therefore be represented as follows:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is </w:t>
  </w:r>
  <w:del w:author="Cooper W.">
    <w:r>
      <w:delText>deleted text</w:delText>
    </w:r>
  </w:del>
</w:p>
```

The deleted text is contained in a delText node, while the regular text is contained in a t node. *end example*]

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules.
	
	[*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`  <w:t>  significant whitespace  </w:t>`
	`</w:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*] The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

17.3.3.8 dirty (Invalidated Field Cache)

This element specifies that the field has been changed and the results shall be updated on open in a conforming consumer.

This element's content model is defined by the common boolean property definition in §17.17.4.

17.3.3.9 drawing (DrawingML Object)

This element specifies that a DrawingML object is located at this position in the run's contents. The layout properties of this DrawingML object are specified using the WordprocessingML Drawing syntax (§20.4).

[*Example*: Consider a run which consists of a picture which is in line with the text in that paragraph (i.e. on the line and affects the line height). That run would be specified using the following WordprocessingML:

```
<w:r>
  <w:drawing>
    <wp:inline>
      …
    </wp:inline>
  </w:drawing>
</w:r>
```

The drawing element indicates that a DrawingML object and its WordprocessingML Drawing positioning data are located at the current position in the run (e.g. a picture or a chart). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.1. *end note*]

17.3.3.10 hps (Phonetic Guide Text Font Size)

This element specifies the font size which shall be applied to the phonetic guide text in the contents of this run when displayed.

If this element disagrees with the run properties on the phonetic guide text rt element (§17.3.3.24), then those properties shall be ignored and this element shall determine the size of the phonetic guide text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a run of phonetic guide text which must have an explicit font size of 13.5 points. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:hps w:val="27"/>
  …
</w:rubyPr>
```

The hps property is 27 half-points for the ruby text in this run, so the phonetic guide text is displayed in 13.5 point font size. *end example*]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.3.11 hpsBaseText (Phonetic Guide Base Text Font Size)

This element specifies the font size which shall be applied to the base text of this phonetic guide text when displayed. If this element disagrees with the run properties on the phonetic guide base text rubyBase element (§17.3.3.27), then this property shall be ignored and the sz element (§17.3.2.38) in that run shall determine the size of the phonetic guide base text.

[*Example*: Consider a run of phonetic guide base text which shall have an explicit font size of 30 points. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:hpsBaseText w:val="60"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    …
</w:rubyPr>
```

The hpsBaseText property is 60 half-points for the base text in this phonetic guide, so the phonetic guide base text is displayed in 30 point font size. *end example*]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.3.12 hpsRaise (Distance Between Phonetic Guide Text and Phonetic Guide Base Text)

This element specifies the distance which shall be left between the phonetic guide base text and the phonetic guide text when this phonetic guide text is displayed.

[*Example*: Consider a run of phonetic guide text which must have 10 points between the phonetic guide base text and the phonetic guide text. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:hpsRaise w:val="20"/>
  …
</w:rubyPr>
```

The hpsRaise property is 20 half-points for the phonetic guide, so the phonetic guide text is displayed 10 points above the phonetic guide base text. *end example*]

Attributes	Description
val (Half Point	Specifies a positive measurement specified in half-points (1/144 of an inch).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Measurement)	The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.3.3.13 lastRenderedPageBreak (Position of Last Calculated Page Break)

This element specifies that this position delimited the end of a page when this document was last saved by an application which paginates its content.

[*Guidance*: This element must be used by applications to specify the locations of page breaks within a document when it is saved as WordprocessingML, in order to allow other applications (e.g. assistive software) to utilize this information when reading the document. *end guidance*]

[*Example*: Consider a run which consists of the text `This is the end of the page`, where the word end was the last word on a page. If the application saving this file had paginated this content, that information can be saved with the file as follows:

```
<w:r>
  <w:t>This is the end</w:t>
  <w:lastRenderedPageBreak/>
  <w:t xml:space="preserve"> of the page</w:t>
</w:r>
```

The lastRenderedPageBreak element indicates that there was a page break resulting from pagination of this content, which occurred between the word end and the word of. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.14 lid (Language ID for Phonetic Guide)

This element specifies the language which shall be used for this phonetic guide.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a run of phonetic guide text which is using Japanese as it language. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:lid w:val="ja-JP"/>
  …
</w:rubyPr>
```

The lid property is `ja-JP` for the phonetic guide, so the phonetic guide is specified to be Japanese. *end example*]

Attributes	Description
val (Language Code)	Specifies an identifier for a specific language. This code is interpreted in the context of the parent XML element. [*Example*: Consider an object which must specify the English(Canada) language. That object would use an identifier of `en-CA` to specify this language. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lang) is located in §A.1. *end note*]

17.3.3.15 monthLong (Date Block - Long Month Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs) . [*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

The long month format date block shall use a date picture of `MMMM`, retrieving the long month format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is a long date: </w:t>
  <w:monthLong />
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This run specifies that a long month format date block shall be placed after the text string literal `This is a long date:` in the document. Assuming that the host application's primary editing language is French (Canada) and today's date is 2006-04-12, this run would be displayed as follows:

```
This is a long date: avril
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.16 monthShort (Date Block - Short Month Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs). [*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

The short month format date block shall use a date picture of MM, retrieving the short month format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is a short date: </w:t>
  <w:monthShort />
</w:r>
```

This run specifies that a short month format date block must be placed after the text string literal `This is a short date:` in the document. Assuming that the host application's primary editing language is English (Canada) and today's date is 2006-04-12, this run would be displayed as follows:

```
This is a short date: 04
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.17 movie (Embedded Video)

This element specifies a location within a document where the specified parent image shall be treated as a static placeholder for an embedded movie. [*Note*: A list of suggested video types in provided in §15.2.17. *end note*]The specified movie file's contents should be displayed when requested at this location in the document. The location of the embedded movie to be displayed when supported shall be specified by the relationship whose Id attribute matches the id attribute on this element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/movie, or is not present, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it can be ignored.

[*Example*: Consider a WordprocessingML document which contains a DrawingML shape holding the static image for a movie:

```
<w:object>
  <w:drawing>
    …
  </w:drawing>
  <w:movie r:id="rIdMovie" />
</w:object>
```

The movie element specifies that the part targeted by the relationship with an ID of `rIdMovie` must be imported at the beginning of the document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="rIdMovie" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/movie" Target="movie.mov" />
  …
</Relationships>
```

The corresponding relationship part item shows that the movie file is located next to the main document and is named `movie.mov`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.3.3.18 noBreakHyphen (Non Breaking Hyphen Character)

This element specifies that a non breaking hyphen character shall be placed at the current location in the run content.

The behavior of a non-breaking hyphen in run content shall be to display using the same glyph as the hyphen-minus character (U+002D), however, without that hyphen being a line breaking position (unlike the hyphen-minus character, which does allow line breaking).

[*Example*: Consider the following sentence in a WordprocessingML document: 'Each citizen has a unique Social Security Number of the form "999-99-9999", where each 9 represents a decimal digit.' The fragment of this sentence involving the string literal might be represented in WordprocessingML, as follows:

```
<w:r>
  <w:t>Number of the form “999-99-9999”, where</w:t>
</w:r>
```

However, consider the case in which, on rendering, the right margin was such that the quoted string is broken across multiple lines with the hyphens being used as possible line breaking points; for example:.

```
Each citizen has a unique Social Security Number of the form “999-99-
9999”, where …
```

If such line breaks are undesirable, those hyphens can be marked as non-breaking, as follows:

```
<w:r>
  <w:t>Number of the form “999</w:t>
</w:r>
<w:r>
  <w:noBreakHyphen />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <w:t>99</w:t>
  </w:r>
  <w:r>
    <w:noBreakHyphen />
    <w:t>9999”, where</w:t>
  </w:r>
```

in which case, for the same margin settings, the rendered result might be like the following:

```
Each citizen has a unique Social Security Number of the form
“999-99-9999”, where …
```

end example]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.19 object (Embedded Object)

This element specifies that an embedded object is located at this position in the run’s contents. The layout properties of this embedded object, as well as an optional static representation, are specified using the drawing element (§17.3.3.9).

[*Example*: Consider a run which consists of an embedded object which is in line with the text in that paragraph (i.e. on the line and affects the line height). That run would be specified using the following WordprocessingML:

```
<w:r>
  <w:object>
    <w:drawing>
      …
    </w:drawing>
  </w:object>
</w:r>
```

The object element indicates that an embedded object and its positioning data are located at the current position in the run (e.g. an embedded object). *end example*]

Attributes	Description
dxaOrig (Original Image Width)	Specifies the original (natural) width of the image representation of the current control within the document. Some vector image formats do not store a native size within their format, and this attribute shall only be used in those cases to store this information, so that the image can be appropriately restored as needed. If this attribute is omitted, then the natural width of the image as stored in its format shall be used. [*Example*: Consider the following WordprocessingML for an embedded object:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:object w:dxaOrig="3360" w:dyaOrig="2520"> … </w:object> The dxaOrig attribute has a value of 3360, which specifies that the image used for the embedded object doesn't store its native width, but that width should be 3360 twentieths of a point. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
dyaOrig (Original Image Height)	Specifies the original (natural) height of the image representation of the current control within the document. Some vector image formats do not store a native size within their format, and this attribute shall only be used in those cases to store this information, so that the image can be appropriately restored as needed. If this attribute is omitted, then the natural height of the image as stored in its format shall be used. [*Example*: Consider the following WordprocessingML for an embedded object: <w:object w:dxaOrig="3360" w:dyaOrig="2520"> … </w:object> The dyaOrig attribute has a value of 2520, which specifies that the image used for the embedded object doesn't store its native height, but that height should be 2520 twentieths of a point. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Object) is located in §A.1. *end note*]

17.3.3.20 objectEmbed (Embedded Object Properties)

This element specifies the visual properties and associated server application of an embedded object.

[*Example*: The following demonstrates a video file embedded in a WordprocessingML document:

```
<w:object … >
  <w:drawing> … </w:drawing>
  <w:objectEmbed drawAspect="content" r:id="rId3" progId="AVIFile"
shapeId="10"/>
</w:object>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
drawAspect (Object Representation)	Specifies how the object is represented visually in the application. [*Example*: `<w:objectEmbed … drawAspect="content"/>` *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_ObjectDrawAspect simple type (§17.18.60).
fieldCodes (Field Switches)	This element specifies the WordprocessingML field switches which shall be stored with an embedded object, using the set of field switches defined by the LINK field, as specified in §17.16.5.32. This element shall specify the exact field switches for the field which represents the object. [*Rationale*: Legacy word processors used fields to represent embedded objects - this element stores the field switches not explicitly defined for embeddings so as not to lose the fidelity of their contents. *end rationale*] [*Example*: `<w:objectEmbed … fieldCodes="\f 0"/>` This embedded object specifies additional LINK field code values of \f 0, which specifies that the embedded object must retain its source formatting (as defined in §17.16.5.32). *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
id (Relationship to Embedded Object Data) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
progId (Object Application)	Specifies the application associated with the object. [*Example*: `<w:objectEmbed … progId="AVIFile"/>` *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
shapeId (Object Shape)	Specifies the shape with which the object is associated. A shape provides the visual placeholder for an object and this attribute is set to the ID of the placeholder shape. [*Example*: `<w:objectEmbed … shapeId="10"/>` *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectEmbed) is located in §A.1. *end note*]

17.3.3.21 objectLink (Linked Object Properties)

This element specifies the visual properties, associated server application and refresh mode of an embedded linked object.

[*Example*: The following demonstrates a video file embedded in a WordprocessingML document:

```
<w:object … >
  <w:drawing> … </w:drawing>
  <w:objectLink drawAspect="icon" r:id="rId3" progId="AVIFile" shapeId="10"
updateMode="user"/>
</w:object>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

Attributes	Description
drawAspect (Object Representation)	Specifies how the object is represented visually in the application. [*Example*: `<w:objectEmbed … drawAspect="content"/>` *end example*] The possible values for this attribute are defined by the ST_ObjectDrawAspect simple type (§17.18.60).
fieldCodes (Field Switches)	This element specifies the WordprocessingML field switches which shall be stored with an embedded object, using the set of field switches defined by the LINK field, as specified in §17.16.5.32. This element shall specify the exact field switches for the field which represents the object. [*Rationale*: Legacy word processors used fields to represent embedded objects - this element stores the field switches not explicitly defined for embeddings so as not to use the fidelity of their contents. *end rationale*] [*Example*: `<w:objectEmbed … fieldCodes="\f 0"/>` This embedded object specifies additional LINK field code values of \f 0, which specifies that the embedded object must retain its source formatting (as defined in §17.16.5.32). *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
id (Relationship to Embedded Object Data) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.8.2.1).
lockedField (Object Refresh Flag)	Specifies whether the object's appearance is locked. If it is locked, the object's current representation shall be locked to prevent any user interaction or automatic application behavior from modifying its contents. [*Example*: `<w:objectLink … lockedField="true"/>` *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
progId (Object Application)	Specifies the application associated with the object. [*Example*: `<w:objectEmbed … progId="AVIFile"/>` *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
shapeId (Object Shape)	Specifies the shape with which the object is associated. A shape provides the visual placeholder for an object and this attribute is set to the ID of the placeholder shape. [*Example*: `<w:objectEmbed … shapeId="10"/>` *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
updateMode (Object Update Mode)	Specifies how the object is updated with new data - automatically or on-demand by the user. [*Example*: `<w:objectLink … updateMode="always"/>` *end example*] The possible values for this attribute are defined by the ST_ObjectUpdateMode simple

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	type (§17.18.61).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectLink) is located in §A.1. *end note*]

17.3.3.22 pgNum (Page Number Block)

This element specifies the presence of a page number block at the current location in the run content. A *page number block* is a non-editable region of text which shall display the current page using ascending decimal numbers. [*Note*: The page number block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the PAGENUM field is used in its place. *end note*]

A page number block shall be displayed using ascending decimal numbers, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is the current page: </w:t>
  <w:pgNum />
</w:r>
```

This run specifies that a page number block must be placed after the text string literal `This is the current page:` in the document. Assuming that this content is on the first page, this run would be displayed as follows:

```
This is the current page: 1
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.23 ptab (Absolute Position Tab Character)

This element specifies that an absolute position tab character shall be placed at the current location in the run content. An *absolute position tab* is a character which is used to advance the position on the current line of text when displaying this WordprocessingML content, using the following logic:

Regardless of any number of custom tab stops defined using the tabs element (§17.3.1.38) , the absolute position tab character shall advance to the position specified by its alignment and relativeTo attributes. The resulting end position of the tab character shall not be affected by the addition of any custom tab stops or changes to the value of the defaultTabStop element (§17.15.1.25).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If the alignment location specified by the positional tab cannot be found on the current line, because the starting location is past that point, then the tab character shall advance to that location on the next available line in the document.

[*Example*: Consider a paragraph which contains two custom tab stops at 1.5" and 3.5", respectively. These two tab stops would be contained within a tabs element defining the set of tab stops of the paragraph as follows:

```
<w:pPr>
  <w:tabs>
    <w:tab w:val="start" w:pos="2160" />
    <w:tab w:val="start" w:pos="5040" />
  </w:tabs>
</w:pPr>
```

If a positional tab character was added to a run in this paragraph starting at 1” inside the margin and was defined as follows:

```
<w:ptab … w:alignment="center" w:relativeTo="margin" />
```

This positional tab would then ignore the next custom tab stop and the indents on the current paragraph defined using the ind element (§17.3.1.12) and would advance to the center of the line with respect to the text margins, moving to a new line if needed. *end example*]

Attributes	Description
alignment (Positional Tab Stop Alignment)	Specifies the location of the positional tab stop on the line, as well as the alignment which shall be applied to text subsequent to the current positional tab stop. [*Example*: Consider a positional tab stop in a WordprocessingML document who must move to the left edge of the text margins and whose subsequent text should be left aligned. This positional tab stop would be defined as follows: `<w:ptab w:alignment="left" w:relativeTo="margin" … />` The alignment attribute specifies that this absolute position tab stop must align on the left edge of the line relative to the margin. *end example*] The possible values for this attribute are defined by the ST_PTabAlignment simple type (§17.18.71).
leader (Tab Leader Character)	Specifies the character which shall be used to fill in the space created by a positional tab. This character shall be repeated as required to completely fill the tab spacing generated by the positional tab character. [*Example*: Consider a positional tab stop which should be preceded by a sequence of underscore characters, as follows: _____Text at the positional tab stop

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This tab stop would have a leader attribute value of underscore, indicating that the tab stop must be preceded by underscore characters as needed to fill the tab spacing. *end example*] The possible values for this attribute are defined by the ST_PTabLeader simple type (§17.18.72).
relativeTo (Positional Tab Base)	Specifies the extents which shall be used to calculate the absolute positioning of this positional tab character. [*Example*: Consider a positional tab stop in a WordprocessingML document that should have a resulting position that is centered on the text margins, ignoring both any custom tab stops and any text indents on the paragraph. This positional tab stop would be defined as follows: `<w:ptab w:relativeTo="margin" … />` The relativeTo attribute specifies that this absolute position tab stop must be relative to the margin. *end example*] The possible values for this attribute are defined by the ST_PTabRelativeTo simple type (§17.18.73).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PTab) is located in §A.1. *end note*]

17.3.3.24 rt (Phonetic Guide Text)

This element specifies the presence of the guide text within a phonetic guide at the current location in the document.

The contents of the guide text run are specified in the child r element (§17.3.2.25).

[*Example*: Consider the following two runs, each containing a phonetic guide:

t ōkyō
東京

The guide text run would be specified using the following WordprocessingML:

```
<w:rt>
  …
  <w:r>
    <w:t>tō</w:t>
  </w:r>
</w:rt>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The guide text is contained in a run within the rt element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RubyContent) is located in §A.1. *end note*]

17.3.3.25 ruby (Phonetic Guide)

This element specifies the presence of a phonetic guide at the current location in the document. A *phonetic guide* (often called ruby text) is a run of content with base text which appears at the normal baseline location for text in this run, with phonetic guide text displayed above it in the document. The resulting construct is called a phonetic guide as it is typically used to map words in one language to another phonetically.

The base text is stored in the rubyBase element (§17.3.3.27) and the guide text is stored in the rt element (§17.3.3.24).

[*Example*: Consider the following two runs, each containing a phonetic guide:

tōkyō
東京

The first run would be specified using the following WordprocessingML:

```
<w:r>
  <w:ruby>
    <w:rubyPr>
      …
    </w:rubyPr>
    <w:rt>
      …
      <w:r>
        <w:t>tō</w:t>
      </w:r>
    </w:rt>
    <w:rubyBase>
      …
      <w:r>
        <w:t>東</w:t>
      </w:r>
    </w:rubyBase>
  </w:ruby>
</w:r>
```

The base text is contained in a run within the rubyBase element, and the guide text is contained in a run within the rt element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ruby) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.3.3.26 rubyAlign (Phonetic Guide Text Alignment)

This element specifies the alignment setting which shall be used to determine the placement of phonetic guide text with respect to the base text when this phonetic guide is displayed.

[*Example*: Consider a run of phonetic guide text which must have the ruby text positioned to the far left of the base text. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:rubyAlign w:val="left"/>
  …
</w:rubyPr>
```

The rubyAlign property is `left` for the phonetic guide, so the ruby text is displayed on the left side of the base text. *end example*]

Attributes	Description

val (Phonetic Guide Text Alignment Value)	Specifies the type of alignment to be applied to the phonetic guide text.
	
	[*Example*: Consider a run of phonetic guide text which must have the ruby text positioned to the far right of the base text. This constraint is specified using the following WordprocessingML:
	
	`<w:rubyPr>`
	`  …`
	`  <w:rubyAlign w:val="left"/>`
	`  …`
	`</w:rubyPr>`
	
	The value of the val attribute is `right` for the phonetic guide, so the ruby text is displayed on the right side of the base text. *end example*]
	
	The possible values for this attribute are defined by the ST_RubyAlign simple type (§17.18.75).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RubyAlign) is located in §A.1. *end note*]

17.3.3.27 rubyBase (Phonetic Guide Base Text)

This element specifies the presence of the base text within a phonetic guide at the current location in the document.

The contents of the base text run are specified in the child r element (§17.3.2.25).

[*Example*: Consider the following two runs, each containing a phonetic guide:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

tōkyō
東京

The base text run would be specified using the following WordprocessingML:

```
<w:rubyBase>
  …
  <w:r>
    <w:t>東/w:t>
  </w:r>
</w:rubyBase>
```

The base text is contained in a run within the rubyBase element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RubyContent) is located in §A.1. *end note*]

17.3.3.28 rubyPr (Phonetic Guide Properties)

This element specifies a set of properties which determine the behavior and appearance of a phonetic guide within the document.

[*Example*: Consider the following two runs, each containing a phonetic guide:

tōkyō
東京

The properties for both of these phonetic guides are as follows:

```
<w:r>
  <w:ruby>
    <w:rubyPr>
      <w:rubyAlign w:val="distributeSpace" />
      <w:hps w:val="16" />
      <w:hpsRaise w:val="20" />
      <w:hpsBaseText w:val="22" />
      <w:lid w:val="ja-JP" />
    </w:rubyPr>
    …
  </w:ruby>
</w:r>
```

The phonetic guide properties specify that the guide text must be:

- Distributed across the top (using the rubyAlign element)
- 8 point font face (using the hps element)
- 10 points above the base text (using the hpaRaise element)
- Japanese (using the lid element)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

As well, the phonetic guide properties specify that the base text must be:

- 11 point font face (using the hpsBaseText element)

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RubyPr) is located in §A.1. *end note*]

17.3.3.29 softHyphen (Optional Hyphen Character)

This element specifies that an optional hyphen character shall be placed at the current location in the run content. An *optional hyphen* is a character which can be used as a line breaking character for the current line of text when displaying this WordprocessingML content, using the following logic:

- When this character is not the character which is used to break the line, then it shall not change the normal display of text (it shall have zero width)
- When this character is the character used to break the line, it shall display using the hyphen-minus character within the display of text

[*Note*: This character is typically used to mark locations where a word can optionally be hyphenated without causing the hyphen character to be displayed unnecessarily. *end note*]

[*Example*: Consider the following sentence in a WordprocessingML document:

```
This sentence needs to be long enough to cause some kind of line breaking.
```

Normally, just as shown above, this sentence not would be displayed on a single line as it is long enough to require line breaking (given the width of the current page). However, if an optional hyphen were inserted after the letter r in `breaking`, as follows:

```
<w:r>
  <w:t>This sentence needs to be long enough to cause some kind of line br</w:t>
  <w:softHyphen/>
  <w:t>eaking.</w:t>
</w:r>
```

This would allow a break at that position, and when that location is the point of the line break, would insert a hyphen-minus in the word after that character:

```
This sentence needs to be long enough to cause some kind of line breaking.
```

If this was not the point of the line break, then no character would be displayed at that location:

```
This sentence should not be long enough to cause line breaking.
```

The sentence now does not break at that location, so no hyphen appears in the word `breaking`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.30 sym (Symbol Character)

This element specifies the presence of a symbol character at the current location in the run's content. A *symbol character* is a special character within a run's content which does not use any of the run fonts specified in the rFonts element (§17.3.2.26) (or by the style hierarchy).

Instead, this character shall be determined by pulling the character with the hexadecimal value specified in the char attribute from the font specified in the font attribute.

[*Example*: Consider a run containing the following run content:

```
This is a symbol character:💻
```

The last character in that run is a symbol character from the Wingdings font, and the run is specified as follows:

```
<w:r>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:hAnsi="Courier New" />
  </w:rPr>
  <w:t>This is a symbol character:</w:t>
  <w:sym w:font="Wingdings" w:char="F03A" />
</w:r>
```

The resulting symbol is the specified using the sym element, and consists of character code `003A` formatted as `Wingdings`, even though the run properties specify the `Courier New` font. *end example*]

Attributes	Description
char (Symbol Character Code)	Specifies the hexadecimal code for the Unicode character value of the symbol. When this value is stored in the char attribute, it can be stored in either of the following two formats: Directly in its Unicode character value from the font glyph In a Unicode character value created by adding F000 to the actual character value, shifting the character value of this character into the Unicode private use area. [*Note*: The use of the latter syntax allows for interoperability with legacy word processing formats, as they used this technique to store the fact that a particular character or set of characters came from a font which was not Unicode compliant, and therefore any font matching performed on this range (if the specified font was not present) would be undesirable, as the resulting glyphs and their appearance could not be predicted. *end note*] [*Example*: Consider a run with a single symbol character defined as follows: `<w:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:rPr> <w:rFonts w:ascii="Arial Black" w:hAnsi="Arial Black" /> </w:rPr> <w:sym w:font="Wingdings" w:char="F045" /> </w:r> The symbol character must use the font defined in its font attribute and hence use the Wingdings font. The character value for the character to be used from this font is obtained by removing the F000 value from the value in the char attribute, and therefore is the character at hexadecimal position 0045 in that font. *end example*] The possible values for this attribute are defined by the ST_ShortHexNumber simple type (§17.18.79).
font (Symbol Character Font)	Specifies a font which shall be used to format this symbol character. [*Example*: Consider a run with a single symbol character defined as follows: <w:r> <w:rPr> <w:rFonts w:ascii="Arial Black" w:hAnsi="Arial Black" /> </w:rPr> <w:sym w:font="Wingdings" w:char="F045" /> </w:r> Although the run specifies that its contents must use the Arial Black font, the symbol character must use the font defined in its font attribute and hence use the Wingdings font. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sym) is located in §A.1. *end note*]

17.3.3.31 t (Text)

This element specifies that this run contains literal text which shall be displayed in the document. The t element shall be used for all text runs which are not:

- Part of a region of text that is contained in a deleted region using the del element (§17.13.5.14)
- Part of a region of text that is contained within a field code

[*Example*: Consider a paragraph of WordprocessingML content which reads This is text. This paragraph would therefore be represented as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t>This is text</w:t>
  </w:r>
</w:p>
```

The text is contained in a t node. *end example*]

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules.
	
	[*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`  <w:t>  significant whitespace  </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

17.3.3.32 tab (Tab Character)

This element specifies that a tab character shall be placed at the current location in the run content. An *tab* is a character which is used to advance the position on the current line of text when displaying this WordprocessingML content, using the following logic:

- When there are one or more custom tab stops defined using the tabs element (§17.3.1.38) , then the tab character shall advance to the next custom tab stop location which is further along than the starting location of the tab
- When there are no custom tab stops which are further than the current position in the line, the tab character shall advance to the nearest multiple of the defaultTabStop element (§17.15.1.25) width value.

[*Example*: Consider a paragraph which contains two custom tab stops at 1.5" and 3.5", respectively. These two tab stops would be contained within a tabs element defining the set of tab stops of the paragraph as follows:

```
<w:pPr>
  <w:tabs>
    <w:tab w:val="start" w:pos="2160" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <w:tab w:val="start" w:pos="5040" />
    </w:tabs>
  </w:pPr>
```

If a tab character was added to a run in this paragraph and appeared 1.4” along the line after all preceding content was laid out, then this tab would move the position to 1.5”. If the tab character appeared 1.6” along the line after all preceding content was laid out, then this tab would move the position to 3.5”. In both cases, the tab advanced to the next custom tab stop. *end example*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.33 yearLong (Date Block - Long Year Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs). [*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run’s lang property (§17.3.2.20).

The long year format date block shall use a date picture of YYYY, retrieving the long year format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is a long date: </w:t>
  <w:yearLong />
</w:r>
```

This run specifies that a long year format date block must be placed after the text string literal `This is a long date:` in the document. Assuming that the host application’s primary editing language is English (Canada) and today’s date is 2006-04-12, this run would be displayed as follows:

```
This is a long date: 2006
```

end example]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.1. *end note*]

17.3.3.34 yearShort (Date Block - Short Year Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs). [*Note*: The date block is a legacy construct used for compatibility with older word

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's lang property (§17.3.2.20).

The short year format date block shall use a date picture of YY, retrieving the short year format for the primary editing language.

[*Example*: Consider a WordprocessingML run with the following run content:

```
<w:r>
  <w:t xml:space="preserve">This is a short date: </w:t>
  <w:yearShort />
</w:r>
```

This run specifies that a short year format date block shall be placed after the text string literal `This is a short date:` in the document. Assuming that the host application's primary editing language is French (Canada) and today's date is 2006-04-12, this run would be displayed as follows:

```
This is a short date: 06
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.3.4 Border Properties (CT_Border)

This common complex type specifies the set of attributes used to define a object's border.

[*Example*: Consider the following run border:

```
<w:r>
  <w:rPr>
    <w:bdr w:val="single" w:sz="36" w:space="0" w:themeColor="accent1"
w:themeTint="66" />
  </w:rPr>
  <w:t xml:space="preserve">run one</w:t>
</w:r>
```

The bdr element specifies a single line border with a width of 3.5 points using the document's `accent1` theme color. *end example*]

Attributes	Description
color (Border Color)	Specifies the color for this border. This value can be defined as either:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	• A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*] . RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the border color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*] [*Example*: Consider a border color with value `auto`, as follows: `<w:bottom … w:color="auto"/>` This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*] If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
frame (Create Frame Effect)	Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text. If this attribute is omitted, then the border is not given any frame effect. [*Example*: Consider a bottom border which must appear with a frame effect, which is specified in the following WordprocessingML: `<w:bottom w:frame="true" … />` This frame's val is `true`, indicating that the border frame effect must be applied. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
shadow (Border Shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following WordprocessingML: `<w:bottom w:shadow="true" … />` This frame's val is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

space (Border Spacing Measurement)	Specifies the spacing offset that shall be used to place this border on the parent object.
	
	When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§17.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points.
	
	When a document has a page border that is relative to the text extents (using a value of `text` in the offsetFrom attribute on pgBorders (§17.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.
	
	[*Example*: Consider a document with a set of page borders all specified to appear 24 points from the edge of the page. The resulting WordprocessingML would be as follows:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:bottom … w:space="24" />`
	`</w:pgBorders>`
	
	The offsetFrom attribute specifies that the space value provides the offset of the page border from the page edge, and the value of the space attribute specifies that the page offset must be 24 points. *end example*]
	
	The possible values for this attribute are defined by the ST_PointMeasure simple type (§17.18.68).

sz (Border Width)	Specifies the width of the current border. If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value. If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range can be reassigned to a more appropriate value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<w:top w:val="dashed" w:sz="24" …/>`
	`<w:left w:val="dashed" w:sz="24" …/>`
	`<w:bottom w:val="dashed" w:sz="24" …/>`
	`<w:right w:val="dashed" w:sz="24" …/>`
	
	The border style is specified using the val attribute, and because that border style is a line border (dashed), the sz attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The default value of this attribute is application-defined.
	[*Note*: It is recommended that implementers write this attribute to ensure interoperability. *end note*]
	
	The possible values for this attribute are defined by the ST_EighthPointMeasure simple type (§17.18.23).

themeColor (Border Theme Color)	Specifies the base theme color used to generate the border color. The border color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to be set centrally in the document.
	
	To determine the color to display, the following actions are performed:
	• Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read.
	• Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color.
	• The specified color is modified based on the presence of the themeTint or themeShade attribute.
	
	[*Example*: Consider a set of borders configured to use the accent2 theme color, resulting in the following WordprocessingML markup:
	
	`<w:top … w:themeColor="accent2" w:themeTint="99" />`
	`<w:bottom … w:themeColor="accent2" w:themeTint="99" />`
	`<w:left … w:themeColor="accent2" w:themeTint="99" />`
	`<w:right … w:themeColor="accent2" w:themeTint="99" />`
	
	If the Settings part contained the following markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:clrSchemeMapping … w:accent2="accent2"/> and the Theme part contained the following XML markup: <a:accent2> <a:srgbClr val="4F81BD"/> </a:accent2> the resulting border color would be 95B3D7 (the result of a 60% tint applied to the original theme color; see the calculations in themeTint below for details). *end example*] The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).
themeShade (Border Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border. [*Example*: Consider a shade of 40% applied to a border in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * Shade_{percentage}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is C0504D.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$and converting back to RGB, we get 943634. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Border Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 \\ = 153 \\ = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L'} = \mathrm{L} * \mathrm{Tint_{pct}} + \left(1 - \mathrm{Tint_{pct}}\right)$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `4F81BD`. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6)$$ $$= 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get `95B3D7`. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Border Style)	Specifies the style of border used on this object. This border can either be an art border (a repeated image along the borders - shall only be used for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<w:left w:val="single" …/>` This border's val is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_Border simple type (§17.18.2).

[*Note*: The W3C XML Schema definition of this complex type's content model (CT_Border) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.3.5 Shading Properties (CT_Shd)

This common complex type specifies the set of attributes used to define a object's shading.

[*Example*: Consider the following paragraph shading:

```
<w:pPr>
  <w:shd w:val="pct20" w:themeColor="accent6" w:themeFill="accent3" />
</w:pPr>
```

The resulting paragraph uses the background color `accent3` under the foreground pattern color `accent6` as specified by the `pct20` pattern mask. *end example*]

Attributes	Description
color (Shading Pattern Color)	Specifies the color used for any foreground pattern specified for this shading using the val attribute. This color can either be presented as a hex value (in RRGGBB format), or `auto` to allow a consumer to automatically determine the foreground shading color as appropriate. If the shading style (the val attribute) specifies the use of no shading format or is omitted, then this property has no effect. Also, if the shading specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value. If this attribute is omitted, then its value shall be assumed to be `auto`. [*Example*: Consider a shading of style `pct20` with a foreground color value of `auto`, as follows: `<w:shd w:val="pct20"… w:color="auto"/>` The foreground color for this shading pattern therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the shading color can be distinguished against the page's background color. *end example*] The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
fill (Shading Background Color)	Specifies the color used for the background for this shading. This color can either be presented as a hex value (in RRGGBB format), or `auto` to allow a consumer to automatically determine the background shading color as appropriate. If this attribute is omitted, then its value shall be assumed to be `auto`. [*Example*: Consider a shading using a background color of hex value C3D69B, using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:shd w:val="pct15" w:fill="C3D69B" /> The background color for this shading therefore is a color with a hex value of C3D69B. *end example*] If the shading specifies the use of a theme color via the themeFill attribute, then this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
themeColor (Shading Pattern Theme Color)	Specifies a theme color which should be applied to any foreground pattern specified for this shading using the val attribute. The specified theme color is a reference to one of the predefined theme colors, located in the document's themes part, which allows for color information to be set centrally in the document. If this attribute is omitted, then no theme color is applied, and the color attribute shall be used to determine the shading pattern color. [*Example*: Consider a paragraph which must have a background consisting of a theme color accent3 with a theme color accent6 overlaid using a 20% fill pattern. This requirement is specified using the following WordprocessingML: <w:pPr> <w:shd w:val="pct20" w:themeColor="accent6" w:themeFill="accent3" /> </w:pPr> The resulting paragraph uses the foreground pattern color accent6 in the region specified by the pct20 pattern mask. *end example*] The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).
themeFill (Shading Background Theme Color)	Specifies a theme color which should be applied to the background for this shading. The specified theme color is a reference to one of the predefined theme colors, located in the document's themes part, which allows for color information to be set centrally in the document. If this attribute is omitted, then no theme color is applied, and the fill attribute shall be used to determine the shading background color. [*Example*: Consider a paragraph which must have a background consisting of a theme

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	color `accent3` with a theme color `accent6` overlaid using a 20% fill pattern. This requirement is specified using the following WordprocessingML: `<w:shd w:val="pct20" w:themeColor="accent6" w:themeFill="accent3" />` The resulting shading uses the background color specified by the `accent3` theme color. *end example*] The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).
themeFillShade (Shading Background Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this shading color. If the themeFillShade is supplied, then it is applied to the RGB value of the themeFill color (from the theme part) to determine the final color applied to this border. The themeFillShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border. [*Example*: Consider a shade of 40% applied to a background shading color in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeFillShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Shade}_{\mathrm{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	$$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get 943634. This transformed value can be seen in the resulting shading's fill attribute: `<w:shd w:val="pct15" w:fill="943634" w:themeFill="accent2" w:themeFillShade="BF" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeFillTint (Shading Background Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this shading instance. If the themeFillTint is supplied, then it is applied to the RGB value of the themeFill color (from the theme part) to determine the final color applied to this border. The themeFillTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 \\ = 153 \\ = 99(hex)$$ The resulting themeFillTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L'} = \mathrm{L} * \mathrm{Tint_{pct}} + \left(1 - \mathrm{Tint_{pct}}\right)$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is 4F81BD.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6)$$ $$= 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get 95B3D7. This transformed value can be seen in the resulting shading's fill attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:fill="95B3D7" w:themeFillColor="accent2" w:themeFillTint="99" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeShade (Shading Pattern Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this shading color. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the themeColor color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border. [*Example*: Consider a shade of 40% applied to a background shading color in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255$$ $$= 102$$ $$= 66(hex)$$ Te resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	$$L' = L * \text{Shade}_{\text{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get `943634`. This transformed value can be seen in the resulting background's color attribute: `<w:shd w:val="pct15" w:color="943634" w:themeColor="accent2" w:themeShade="BF" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Shading Pattern Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this shading instance. If the themeTint is supplied, then it is applied to the RGB value of the themeColor color (from the theme part) to determine the final color applied to this border. The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 = 153 = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * \text{Tint}_{pct} + (1 - \text{Tint}_{pct})$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `4F81BD`. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6) = 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get `95B3D7`. This transformed value can be seen in the resulting shading's color attribute: `<w:shd w:val="pct15" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Shading Pattern)	Specifies the pattern which shall be used to lay the pattern color over the background color for this paragraph shading. This pattern consists of a mask which is applied over the background shading color to get the locations where the pattern color should be shown. Each of these possible masks are shown in the simple type values referenced below. [*Example*: Consider a shaded paragraph which uses a 10 percent foreground fill, resulting in the following WordprocessingML: `<w:shd w:val="pct10" … />` This shading val is `pct10`, indicating that the border style is a 10 percent foreground fill mask. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Shd simple type (§17.18.78).

[*Note*: The W3C XML Schema definition of this complex type's content model (CT_Shd) is located in §A.1. *end note*]

17.4 Tables

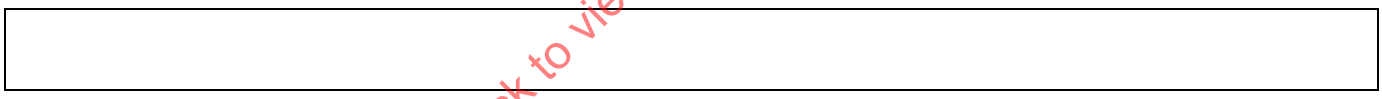
Another form of block-level content in WordprocessingML, a *table* is a set of paragraphs (and other block-level content) arranged in *rows* and *columns*. Tables in WordprocessingML are defined via the tbl element, which is analogous to the HTML `<table>` tag. The table element specifies the location of a table present in the document.

A tbl element (§17.4.37) has two elements that define its properties:

- tblPr (§17.4.59), which defines the set of table-wide properties (such as style and width)
- tblGrid (§17.4.48), which defines the grid layout of the table.

A tbl element can also contain an arbitrary non-zero number of rows, where each row is specified with a tr element (§17.4.78). Each tr element can contain an arbitrary non-zero number of cells, where each cell is specified with a tc element (§17.4.65).

[*Example*: Consider an empty one-cell table (i.e.; a table with one row, one column) and 1 point borders on all sides as follows:



This table is represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tblPr>
    <w:tblW w:w="5000" w:type="pct"/>
    <w:tblBorders>
      <w:top w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:start w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:bottom w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:end w:val="single" w:sz="4" w:space="0" w:color="auto"/>
    </w:tblBorders>
  </w:tblPr>
  <w:tblGrid>
    <w:gridCol w:w="10296"/>
  </w:tblGrid>
  <w:tr>
    <w:tc>
      <w:tcPr>
        <w:tcW w:w="0" w:type="auto"/>
      </w:tcPr>
      <w:p/>
    </w:tc>
  </w:tr>
</w:tbl>
```

This table specifies table-wide properties of 100% of page width using the tblW element and the set of table borders using the tblBorders element, the table grid which defines a set of shared vertical edges within the table using the tblGrid element, and a single row using the tr element. *end example*]

17.4.1 bidiVisual (Visually Right to Left Table)

This element specifies that the cells with this table shall be visually represented in a right to left direction. This element also affects the application of all table-level properties.

When this property is specified, then the ordering of all cells (and table-level properties) in this table shall be applied to the table assuming that the table is a normal left to right table, but the table cells shall be displayed in a right to left direction. [*Example*: A left border on the first table cell must be displayed on the right side of that cell (which would be the rightmost cell) in a visually right to left table. *end example*]

If this element is omitted, then the table shall not be presented right to left.

[*Example*: Consider the following table which has the logical right to left property set:

	Three	Two	One

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This property would be specified in the WordprocessingML as follows:

```
<w:tblPr>
  <w:bidiVisual/>
</w:tblPr>
```

Since the bidiVisual element specifies this is a visually right to left table, the actual table data would be stored in its logical order as follows:

```
<w:tr>
  <w:tc>
    <w:p>
      <w:r>
        <w:t>One</w:t>
      </w:r>
    </w:p>
  </w:tc>
  <w:tc>
    <w:p>
      <w:r>
        <w:t>Two</w:t>
      </w:r>
    </w:p>
  </w:tc>
  <w:tc>
    <w:p>
      <w:r>
        <w:t>Three</w:t>
      </w:r>
    </w:p>
  </w:tc>
  <w:tc>
    <w:p/>
  </w:tc>
</w:tr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The first logical cell with text One is stored first in the file format, and displayed on the rightmost in this table visually. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

©ISO/IEC 2016 – All rights reserved

17.4.2 bottom (Table Cell Bottom Margin Exception)

This element specifies the amount of space which shall be left between the bottom extent of the cell contents and the border of a specific table cell within a table. This setting shall override the table cell bottom margin definition specified by the bottom element contained within the table properties (§17.4.5).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the bottom element contained within the table properties (§17.4.5).

[*Example*: Consider a table with two cells in which the first table cell's bottom margin is specified via an exception to be ten times larger (0.2 inches vs. 0.02 inches) than the other table cell margins:

This text fills the extents of the cell.
So does this

The first cell in the table would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcMar>
      <w:bottom w:w="288" w:type="dxa" />
    </w:tcMar>
  </w:tcPr>
</w:tc>
```

The first cell in this table has an exception applied to the table cell bottom cell margin setting it to 288 twentieths of a point (0.2 inches). *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.3 bottom (Table Cell Bottom Border)

This element specifies the border which shall be displayed at the bottom of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the bottom of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a table in which the first cell in the first row specifies a bottom cell border

R1C1	R1C2
R2C1	R2C2

This bottom cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
      <w:bottom w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The bottom element specifies a three point border of type `thinThinThickSmallGap`. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.4 bottom (Table Bottom Border)

This element specifies the border which shall be displayed at the bottom of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the bottom of this table shall have the border specified by the associated table style. If no bottom border is specified in the style hierarchy, then this table shall not have a bottom border.

[*Example*: Consider a table in which the table properties specifies a bottom table border, as follows:

R1C1	R1C2
R2C1	R2C2

This bottom table border is specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tbl>
  <w:tblPr>
    <w:tblBorders>
      <w:bottom w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="D0D0D0" w:themeColor="accent3" w:themeTint="99"/>
    </w:tblBorders>
  </w:tblPr>
  …
</w:tbl>
```

The bottom element specifies a three point bottom table border of type `thinThinThickMediumGap`. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.5 bottom (Table Cell Bottom Margin Default)

This element specifies the amount of space which shall be left between the bottom extent of the cell contents and the border of all table cells within the parent table (or table row). This setting can be overridden by the table cell bottom margin definition specified by the bottom element contained within the table cell's properties (§17.4.2).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a bottom margin is never specified in the style hierarchy, then this table shall have no bottom cell padding by default (excepting individual cell overrides).

[*Example*: Consider a two by two table in which the default table cell bottom margin is specified to be exactly 0.25 inches, as follows (marked with an arrow in the first table cell below):

R1C1	R2C1
R2C1	R2C2

This table property is specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tblPr>
    <w:tblCellMar>
      <w:bottom w:w="360" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  …
</w:tbl>
```

Every cell in the table has a default cell margin setting it to `360` twentieths of a point. *end example*]

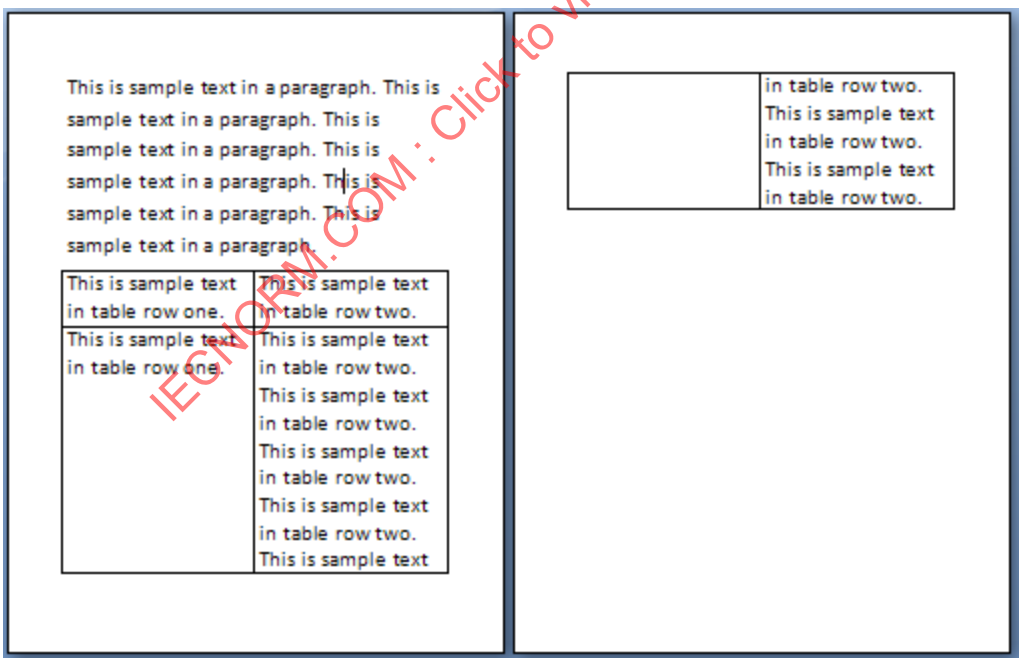
This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.6 cantSplit (Table Row Cannot Break Across Pages)

This element specifies whether the contents within the current cell shall be rendered on a single page. When displaying the contents of a table cell (such as the table cells in ISO/IEC 29500), it is possible that a page break would fall within the contents of a table cell, causing the contents of that cell to be displayed across two different pages. If this property is set, then all contents of a table row shall be rendered on the same page by moving the start of the current row to the start of a new page if necessary. If the contents of this table row cannot fit on a single page, then this row shall start on a new page and flow onto multiple pages as necessary.

If this element is not present, the default behavior is dictated by the setting in the associated table style. If this property is not specified in the style hierarchy, then this table row shall be allowed to split across multiple pages.

[*Example*: Consider the following content displayed on two different pages of a WordprocessingML document:

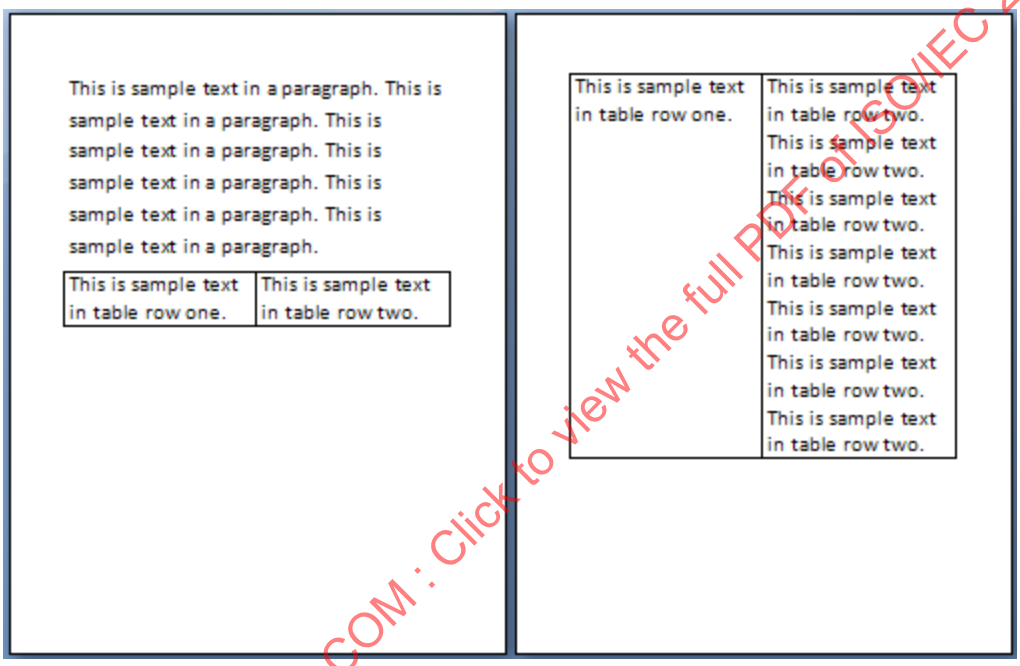


IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

When this content is displayed, the contents of the 2nd table row in this document are displayed across two different pages. If the contents of this row are to be displayed on one page, then this requirement would be specified as follows:

```
<w:tr>
  <w:trPr>
    …
    <w:cantSplit />
  </w:trPr>
  …
</w:tr>
```

The presence of the cantSplit element specifies that the table row must not be broken across multiple pages, therefore the second table row starts on a new page:



This setting therefore ensures that the content is always displayed on a single page (if it fits on one page). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.4.7 cnfStyle (Table Row Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this table row. [*Note*: This property is an optimization which is used by consumers to determine if a given property on a table row is the result of the table style conditional formatting properties vs. direct formatting on the table cell itself. It specifies the components of the conditional formatting in the table style applied to this cell, so that the table's conditional formatting can be applied after the document is displayed without having the table style properties override the style hierarchy. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then its value shall be assumed to be zero for all entries in the bit mask.

[*Example*: Consider a table row in the top of a table with a table style applied. This table cell would need to specify the following WordprocessingML to express that fact:

```
<w:tr>
  <w:trPr>
    <w:cnfStyle w:firstRow="true" />
    …
  </w:trPr>
  …
</w:tr>
```

This table row specifies that it has the conditional properties from the table style for the first row of the parent table by setting the appropriate attribute value. *end example*]

Attributes	Description

evenHBand (Even Numbered Horizontal Band)	Specifies that the object has inherited the conditional properties applied to the even numbered horizontal bands of the parent object.
	
	[*Example*: Consider a paragraph in the second row of a table with a table style applied, and where the band width is one row. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:evenHBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the even numbered horizontal bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

evenVBand (Even Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the even numbered vertical bands of the parent object.
	
	[*Example*: Consider a paragraph in the second column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:evenVBand="true" />`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the even numbered vertical bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstColumn (First Column)	Specifies that the object has inherited the conditional properties applied to the first column of the parent object. [*Example*: Consider a paragraph in the first column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:firstColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRow (First Row)	Specifies that the object has inherited the conditional properties applied to the first row of the parent object. [*Example*: Consider a paragraph in the top row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:firstRow="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the first row of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRowFirstColumn (First Row and	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and first column of the parent object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

First Column)	[*Example*: Consider a paragraph in the first row and first column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" w:firstColumn="true" w:firstRowFirstColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and first column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

firstRowLastColumn (First Row and Last Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and last column of the parent object.
	
	[*Example*: Consider a paragraph in the first row and last column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" w:lastColumn="true" w:firstRowLastColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and last column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

lastColumn (Last Column)	Specifies that the object has inherited the conditional properties applied to the last column of the parent object.
	
	[*Example*: Consider a paragraph in the last column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastColumn="true" />`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the last column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRow (Last Row)	Specifies that the object has inherited the conditional properties applied to the last row of the parent object. [*Example*: Consider a paragraph in the bottom row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastRow="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the last row of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRowFirstColumn (Last Row and First Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the last row and first column of the parent object. [*Example*: Consider a paragraph in the last row and first column of a table. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastRow="true" w:firstColumn="true" w:lastRowFirstColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRowLastColum	Specifies that the object has inherited the conditional properties applied to the cell that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

n (Last Row and Last Column)	is in the last row and last column of the parent object.
	
	[*Example*: Consider a paragraph in the last row and last column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastRow="true" w:lastColumn="true" w:lastRowLastColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and last column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

oddHBand (Odd Numbered Horizontal Band)	Specifies that the object has inherited the conditional properties applied to the odd numbered horizontal bands of the parent object.
	
	[*Example*: Consider a paragraph in the third row of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:oddHBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the odd numbered horizontal bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

oddVBand (Odd Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the odd numbered vertical bands of the parent object.
	
	[*Example*: Consider a paragraph in the third column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:cnfStyle w:oddVBand="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the odd numbered vertical bands of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cnf) is located in §A.1. *end note*]

17.4.8 cnfStyle (Table Cell Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this table cell. [*Note*: This property is an optimization which is used by consumers to determine if a given property on a table cell is the result of the table style conditional formatting properties vs. direct formatting on the table cell itself. It specifies the components of the conditional formatting in the table style applied to this cell, so that the table's conditional formatting can be applied after the document is displayed without having the table style properties override the style hierarchy. *end note*]

If this element is omitted, then its value shall be assumed to be zero for all entries in the bit mask.

[*Example*: Consider a table cell in the top right corner of a table with a table style applied and where the table is formatted as left to right. This table cell would need to specify the following WordprocessingML to express that fact:

```
<w:tc>
  <w:tcPr>
    <w:cnfStyle w:firstRow="true" w:lastColumn="true"
w:firstRowLastColumn="true" />
    …
  </w:tcPr>
  …
</w:tc>
```

This table cell specifies that it has the conditional properties from the table style for the first column, first row, and the top right corner of the parent table by setting the appropriate attributes. *end example*]

Attributes	Description
evenHBand (Even Numbered Horizontal Band)	Specifies that the object has inherited the conditional properties applied to the even numbered horizontal bands of the parent object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a paragraph in the second row of a table with a table style applied, and where the band width is one row. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:evenHBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the even numbered horizontal bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

evenVBand (Even Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the even numbered vertical bands of the parent object.
	
	[*Example*: Consider a paragraph in the second column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:evenVBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the even numbered vertical bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

firstColumn (First Column)	Specifies that the object has inherited the conditional properties applied to the first column of the parent object.
	
	[*Example*: Consider a paragraph in the first column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstColumn="true" />`
	`  …`
	`  </w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </w:p> This paragraph specifies that it has the conditional properties from the table style for the first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRow (First Row)	Specifies that the object has inherited the conditional properties applied to the first row of the parent object. [*Example*: Consider a paragraph in the top row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:firstRow="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the first row of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRowFirstColumn (First Row and First Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and first column of the parent object. [*Example*: Consider a paragraph in the first row and first column of a table. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:firstRow="true" w:firstColumn="true" w:firstRowFirstColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRowLastColumn (First Row and	Specifies that the object has inherited the conditional properties applied to the cell that is in the first row and last column of the parent object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

Last Column)	[*Example*: Consider a paragraph in the first row and last column of a table. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:firstRow="true" w:lastColumn="true" w:firstRowLastColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the cell in the first row and last column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

lastColumn (Last Column)	Specifies that the object has inherited the conditional properties applied to the last column of the parent object.
	
	[*Example*: Consider a paragraph in the last column of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastColumn="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the last column of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

lastRow (Last Row)	Specifies that the object has inherited the conditional properties applied to the last row of the parent object.
	
	[*Example*: Consider a paragraph in the bottom row of a table with a table style applied. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:lastRow="true" />`
	`  …`
	`  </w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </w:p> This paragraph specifies that it has the conditional properties from the table style for the last row of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRowFirstColumn (Last Row and First Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the last row and first column of the parent object. [*Example*: Consider a paragraph in the last row and first column of a table. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastRow="true" w:firstColumn="true" w:lastRowFirstColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and first column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRowLastColumn (Last Row and Last Column)	Specifies that the object has inherited the conditional properties applied to the cell that is in the last row and last column of the parent object. [*Example*: Consider a paragraph in the last row and last column of a table. This paragraph would need to specify the following WordprocessingML: <w:p> <w:pPr> <w:cnfStyle w:lastRow="true" w:lastColumn="true" w:lastRowLastColumn="true" /> … </w:pPr> … </w:p> This paragraph specifies that it has the conditional properties from the table style for the cell in the last row and last column of the parent table. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
oddHBand (Odd	Specifies that the object has inherited the conditional properties applied to the odd

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

Numbered Horizontal Band)	numbered horizontal bands of the parent object.
	
	[*Example*: Consider a paragraph in the third row of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:oddHBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the odd numbered horizontal bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

oddVBand (Odd Numbered Vertical Band)	Specifies that the object has inherited the conditional properties applied to the odd numbered vertical bands of the parent object.
	
	[*Example*: Consider a paragraph in the third column of a table with a table style applied, and where the band width is one column. This paragraph would need to specify the following WordprocessingML:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:cnfStyle w:oddVBand="true" />`
	`  …`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph specifies that it has the conditional properties from the table style for the odd numbered vertical bands of the parent table. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cnf) is located in §A.1. *end note*]

17.4.9 divId (Associated HTML div ID)

This element specifies the HTML div information which is associated with the current table row. This information, stored in the Web Settings part, is used to associate one or more table rows with a particular HTML `div` element. [*Note*: This property is used when saving an HTML document into the WordprocessingML format

©ISO/IEC 2016 – All rights reserved

in order to prevent a loss of all HTML `div` information, so that the document can later be saved back into HTML format and have the stored information replaced, since the HTML `div` can store formatting properties on arbitrary regions. *end note*]

In order to determine the associated HTML `div` properties, the value of the val attribute on this element is used to look up an associated div element (§17.15.2.8) whose id attribute matches this value.

If this table row does not have a divId element present, then this table row shall not have any associated HTML div information. If this element is present, but the val attribute specifies an id value which does not have an associated div element, then this element is ignored.

[*Example*: Consider an HTML document defined as follows:

```
<html>
  <body>
    <div style="…">
      <table>
        <tr>
          <td>R1C1</td>
          …
        </tr>
      </table>
      <p>
        …
      </p>
    </div>
    …
  </body>
</html>
```

This HTML document specifies a div spanning the table and the first paragraph. If this document is saved into WordprocessingML, then both the rows of the table and the paragraph must have a divId which points at the same div information in the web settings part:

```
<w:trPr>
  …
  <w:divId w:val="1102603671"/>
</w:trPr>
```

The val attribute then points at a div element which stores the associated `div` properties:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:divs>
  <w:div w:id="1102603671">
    …
  </w:div>
</w:divs>
```

This specifies that this table's rows are part of a single HTML `div`. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.4.10 end (Table Cell Trailing Margin Exception)

This element specifies the amount of space which shall be present between the trailing extent of the current cell's text contents and the trailing border of a specific individual table cell within a table. This setting shall override the table cell trailing margin definition specified by the end element contained within the table properties (§17.4.11).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If omitted, then this table cell shall use the trailing cell margins defined in the end element contained within the table properties (§17.4.11).

[*Example*: Consider a two row, two column LTR table in which the first table cell in the second row has a trailing margin which is specified via an exception to be 0.5 inches, the region marked with an arrow in the table below:

R1C1	R1C2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

R2C1 ⟷	R2C2

The exception on this cell would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcMar>
      <w:end w:w="720" w:type="dxa" />
    </w:tcMar>
  </w:tcPr>
  …
</w:tc>
```

The R2C1 cell in this table has an exception applied to the table cell trailing cell margin setting it to `720` twentieths of a point (0.5 inches). *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.11 end (Table Cell Trailing Margin Default)

This element specifies the amount of space which shall be present between the trailing extent of the cell contents and the trailing border of all table cells within the parent table (or table row) . This setting can be overridden by the table cell trailing margin definition specified by the `end` element contained within the table cell's properties (§17.4.10).

This value is specified in the units applied via its `type` attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a trailing margin is never specified in the style hierarchy, this table shall have 115 twentieths of a point (0.08 inches) left cell padding by default (excepting individual cell overrides).

[*Example*: Consider a two by two LTR table in which the default table cell trailing margin is specified to be exactly 0.25 inches, as follows (marked with an arrow in the first table cell below):

R1C1 ⟷	R2C1
R2C1	R2C2

This table property is specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tblPr>
    <w:tblCellMar>
      <w:end w:w="360" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  …
</w:tbl>
```

Every cell in the table has a default trailing cell margin setting it to 360 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.12 end (Table Cell Trailing Edge Border)

This element specifies the border which shall be displayed on the trailing edge of the current table cell (right for LTR tables, left for RTL tables). The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the trailing edge of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

[*Example*: Consider an LTR table in which the second cell in the first row specifies a trailing dge cell border as follows:

R1C1	R1C2
R2C1	R2C2

This trailing edge cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
      <w:end w:val="double" w:sz="4" w:space="0" w:color="FF0000" />
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The end element specifies a ½ point border of type double on the trailing edge of the table cell. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.13 end (Table Trailing Edge Border)

This element specifies the border which shall be displayed at the trailing edge of the current table (right for LTR tables, left for RTL tables). The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the trailing edge of this table shall have the border specified by the associated table style. If no trailing edge border is specified in the style hierarchy, then this table shall not have a trailing edge border.

[*Example*: Consider an LTR table in which the table properties specify a trailing edge table border, as follows:

R1C1	R1C2
R2C1	R2C2

This trailing edge table border is specified using the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:tblBorders>
      <w:end w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="D0D0D0" w:themeColor="accent3" w:themeTint="99"/>
    </w:tblBorders>
  </w:tblPr>
  …
</w:tbl>
```

The end element specifies a three point trailing edge table border of type thinThinThickMediumGap. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.14 gridAfter (Grid Columns After Last Cell)

This element specifies the number of grid columns in the parent table's table grid (§17.4.48; §17.4.47) which shall be left after the last cell in the table row.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element conflicts with the remaining size of the document grid after all table cells in this row have been added to the grid, then it shall be ignored. If this element is not specified, then its value shall be assumed to be zero grid units.

[*Example*: Consider a table whose second row ends before the first row by one grid unit:

In this table, the second row leaves one grid unit after its cell contents, which is represented using the following WordprocessingML:

```
<w:tr>
  <w:trPr>
    <w:gridAfter w:val="1" />
    …
  </w:trPr>
  …
</w:tr>
```

The gridAfter element specifies that 1 grid unit remains after the row's contents. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.4.15 gridBefore (Grid Columns Before First Cell)

This element specifies the number of grid columns in the parent table's table grid (§17.4.48; §17.4.47) which must be skipped before the contents of this table row (its table cells) are added to the parent table. [*Note*: This property is used to specify tables whose leading edge (left for left-to-right tables, right for right-to-left tables) does not start at the first grid column (the same shared edge). *end note*]

If this element is omitted, then its value shall be assumed to be zero grid units. If this element's value is larger than the size of the table grid, then the value shall be ignored and the first cell in the row can span the full table grid (i.e. the second cell, if one exists, should start at the last shared edge in the table).

[*Example*: Consider a table whose second row starts after the first grid unit:

In this table, the second row skips one grid unit at the beginning which is represented by the following WordprocessingML:

```
<w:tr>
  <w:trPr>
    <w:gridBefore w:val="1" />
    …
  </w:trPr>
  …
</w:tr>
```

end example]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

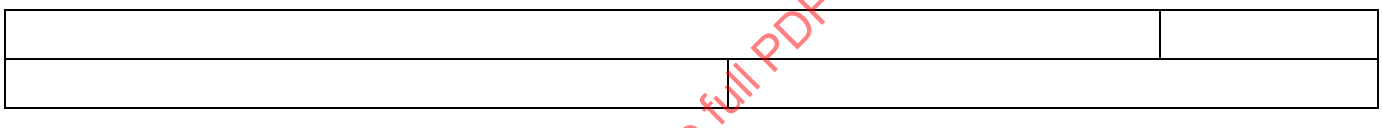
17.4.16 gridCol (Grid Column Definition)

This element specifies the presence and details about a single grid column within a table grid. A *grid column* is a logical column in a table used to specify the presence of a shared vertical edge in the table. When table cells are then added to this table, these shared edges (or grid columns, looking at the column between those shared edges) determine how table cells are placed into the table grid.

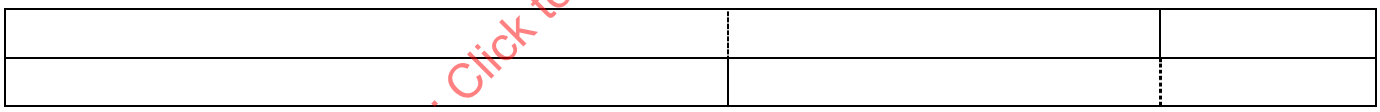
[*Example*: If a table row specifies that it is preceded by two grid columns, then it would start on the third vertical edge in the table including edges which are not shared by all columns. *end example*]

If the table grid does not match the requirements of one or more rows in the table (i.e. it does not define enough grid columns), then the grid can be redefined as needed when the table is processed.

[*Example*: Consider the following, more complex table that has two rows and two columns; as shown, the columns are not aligned:



This table is represented by laying out the cells on a table grid consisting of three table grid columns as follows, each grid column representing a logical vertical column in the table:



The dashed lines represent the virtual vertical continuations of each table grid column, and thus resulting table grid is represented as the following in WordprocessingML:

```
<w:tblGrid>
  <w:gridCol w:w="5051" />
  <w:gridCol w:w="3008" />
  <w:gridCol w:w="1531" />
</w:tblGrid>
```

end example]

Attributes	Description
w (Grid Column	Specifies the width of this grid column.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Width)	[*Note*: This value does not solely determine the actual width of the resulting grid column in the document. When the table is displayed in a document, these widths determine the initial width of each grid column, which can then be overridden by:
	• The table layout algorithm (§17.4.52;§17.4.53) applied to the current table row(s)
	• The preferred widths of specific cells which are part of that grid column as the table is displayed (which is an input to the algorithm above)
	end note]
	
	This value is specified in twentieths of a point.
	
	If this attribute is omitted, then the last saved width of the grid column is assumed to be zero.
	
	[*Example*: Consider the following table grid definition:
	
	`<w:tblGrid>`
	`  <w:gridCol w:w="6888"/>`
	`  <w:gridCol w:w="248"/>`
	`  <w:gridCol w:w="886"/>`
	`  <w:gridCol w:w="1554"/>`
	`</w:tblGrid>`
	
	This table grid specifies four grid columns, each of which has an initial size of 6888 twentieths of a point, 248 twentieths of a point, 886 twentieths of a point, and 1554 twentieths of a point respectively. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblGridCol) is located in §A.1. *end note*]

17.4.17 gridSpan (Grid Columns Spanned by Current Table Cell)

This element specifies the number of grid columns in the parent table's table grid which shall be spanned by the current cell. This property allows cells to have the appearance of being merged, as they span vertical boundaries of other cells in the table.

If this element is omitted, then the number of grid units spanned by this cell shall be assumed to be one. If the number of grid units specified by the val attribute exceeds the size of the table grid, then the table grid shall be augmented as needed to create the number of grid columns required.

[*Example*: Consider the following table that has two rows and two columns where the columns are not aligned:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This table is represented by laying out the cells on a table grid consisting of three table grid columns, each grid column representing a logical vertical column in the table:

The first table cell in the first row spans two grid column units. The second cell in the second row also consumes two grid column units (see the grid lines represented using emphasized lines in the example above). This table is represented using the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:tcPr>
        …
        <w:gridSpan w:val="2" />
      </w:tcPr>
      …
    </w:tc>
    …
  </w:tr>
  <w:tr>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        …
        <w:gridSpan w:val="2" />
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
</w:tbl>
```

The gridSpan element indicates the number of columns spanned by each cell with respect to the table grid (in the case of R1C1 and TR2C2, two. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.4.18 header (Header Cell Reference)

This element specifies a reference, using a unique identifier, to a table header cell that is associated with the current table cell. The identifier representing the reference shall be stored on this element's val attribute and is used to reference the unique identifier value of the *id* attribute of a header cell *tc* element of the current table. The contents of the table header cell *tc* element designated by the specific unique identifier in its *id* attribute shall be used as the table header information associated with the table cell that references that specific unique identifier.

If this element is omitted or the value of the header cell identifier cannot be resolved, no header cell shall be associated with the current table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in WordprocessingML as follows:

```
<w:tbl>
  …
  <w:tr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    …
    <w:tc >
      …
    </w:tc>
    <w:tc w:id="HeaderA">
      …
      <w:p>
        <w:r>
          <w:t>A</w:t>
        </w:r>
      </w:p>
    </w:tc>
    …
  </w:tr>
  <w:tr>
    …
    <w:tc w:id="HeaderC">
      …
      <w:p>
        <w:r>
          <w:t>C</w:t>
        </w:r>
      </w:p>
    </w:tc>
    <w:tc>
      <w:tcPr>
        …
        <w:headers>
          <w:header w:val="HeaderA" />
          <w:header w:val="HeaderC" />
        </w:headers>
        …
      </w:tcPr>
      <w:p>
        <w:r>
          <w:t>x1</w:t>
        </w:r>
      </w:p>
    </w:tc>
    …
  </w:tr>
</w:tbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example the table cell with the content value of x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's styleId.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.4.19 headers (Header Cells Associated With Table Cell)

This element specifies the list of header cells, as specified by children header elements, that provide header information associated with the current table cell. Each header cell shall specify a unique identifier, as specified by the use of the attribute id on the header cell tc element. This element is typically used to gather header information about data and sub header cells.

If this element is omitted or no children header element exists, no header cell shall be associated with the given table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in WordProcessingML as follows:

```
<w:tbl>
  …
  <w:tr>
    …
    <w:tc >
      …
    </w:tc>
    <w:tc w:id="HeaderA">
      …
      <w:p>
        <w:r>
          <w:t>A</w:t>
        </w:r>
      </w:p>
    </w:tc>
    …
  </w:tr>
  <w:tr>
    …
    <w:tc w:id="HeaderC">
      …
      <w:p>
        <w:r>
          <w:t>C</w:t>
        </w:r>
      </w:p>
    </w:tc>
    <w:tc>
      <w:tcPr>
        …
        <w:headers>
          <w:header w:val="HeaderA" />
          <w:header w:val="HeaderC" />
        </w:headers>
        …
      </w:tcPr>
      <w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
          <w:r>
            <w:t>x1</w:t>
          </w:r>
        </w:p>
      </w:tc>
      …
    </w:tr>
  </w:tbl>
```

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Headers) is located in §A.1. *end note*]

17.4.20 hidden (Hidden Table Row Marker)

This element specifies that the glyph representing the end character of current table row shall not be displayed in the current document.

[*Note*: This setting is used to hide the end of row glyph in order to ensure that the entire table row is hidden and not displayed in the document, as if any part of the row is visible, the row is displayed. *end note*]

[*Note*: Applications can have settings which allow hidden content to be displayed, in which case this content can be visible - this property is not meant to supersede that setting. *end note*]

If this element is omitted, then this table row shall not be hidden in the document.

[*Example*: Consider a table with a table row in which the row is specified to be hidden. That requirement is specified using the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:trPr>
      <w:hidden />
      …
    </w:trPr>
    …
  </w:tr>
</w:tbl>
```

In this example this row is not be displayed nor printed, as the hidden element is specified on in table row's properties. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.4.21 hideMark (Ignore End Of Cell Marker In Row Height Calculation)

This element specifies whether the end of cell glyph shall influence the height of the given table row in the table. If it is specified, then only printing characters in this cell shall be used to determine the row height.

[*Rationale*: Typically, the height of a table row is determined by the height of all glyphs in all cells in that row, including the non-printing end of cell glyph characters. However, if these characters are not formatted, they are always created with the document default style properties. This means that the height of a table row cannot ever be reduced below the size of the end of cell marker glyph without manually formatting each paragraph in that run.

In a typical document, this behavior is desirable as it prevents table rows from 'disappearing' if they have no content. However, if a table row is being used as a border (for example, by shading its cells or putting an image in them), then this behavior makes it impossible to have a virtual border that is reasonably small without formatting each cell's content directly. This setting specifies that the end of cell glyph shall be ignored for this cell, allowing it to collapse to the height of its contents without formatting each cell's end of cell marker, which would have the side effect of formatting any text ever entered into that cell. *end rationale*]

If this element is omitted, then the end of cell marker shall be included in the determination of the height of this row.

[*Example*: Consider the following WordprocessingML table:

Here is some small text.			

Notice that the only printing content in this table row is displayed using 5 point font, yet the row height is influenced by the end of cell markers in the empty cells.

If each cell in the second row in this table was set to exclude the table cell from this calculation, using the following WordprocessingML:

```
<w:tcPr>
  <w:hideMark/>
</w:tcPr>
```

The resulting table must exclude the cell markers from the row height calculation:

Here is some text.			

The hideMark element specified that each cell marker was excluded, resulting in the row height being defined by the actual run contents. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This element's content model is defined by the common boolean property definition in §17.17.4.

17.4.22 insideH (Table Inside Horizontal Edges Border)

This element specifies the border which shall be displayed on all horizontal table cell borders which are not on an outmost edge of the parent table (all horizontal borders which are not the topmost or bottommost border). The appearance of this table cell border in the document shall be determined by the following settings:

- The display of the border on interior edges is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the inside horizontal borders of this table shall have the border specified by the associated table style. If no inside horizontal edge border is specified in the style hierarchy, then this table shall not have an inside horizontal edge border.

[*Example*: Consider a table in which the table specifies a border on all interior horizontal and vertical edges, as follows:

R1C1	R1C2
R2C1	R2C2

This interior horizontal cell border is specified using the following WordprocessingML:

```
<w:tblPr>
  <w:tblBorders>
    <w:insideH w:val="thickThinSmallGap" w:sz="24" w:space="0"
      w:color="auto"/>
    <w:insideV w:val="thickThinSmallGap" w:sz="24" w:space="0"
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      w:color="auto"/>
    </w:tblBorders>
    …
  </w:tblPr>
```

The insideH element specifies a 3-point border of type thickThinSmallGap. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.23 insideH (Table Cell Inside Horizontal Edges Border)

This element specifies the border which shall be displayed on all interior horizontal edges of the current group of table cells. [*Note*: Although individual table cells have no concept of an internal edge, which would render this property useless in most cases, it is used to determine the cell borders to apply to a specific group of cells as part of table conditional formatting in a table style, for example, the inside horizontal edges on the set of cells in the first column. *end note*]

The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the specified conditional formatting on the table shall not change the current set of internal edge borders on its set of table cells (i.e. their current setting shall remain unchanged).

[*Example*: Consider a table in which the conditional formatting on the first column specified in the associated table style specifies a double line red cell border for all internal horizontal lines as follows:

R1C1	R1C2
R2C1	R2C2

This inner horizontal edge cell border is specified using the following WordprocessingML:

```
<w:tblStylePr w:type="firstCol">
  <w:tcPr>
    <w:tcBorders>
      <w:insideH w:val="double" w:sz="4" w:space="0" w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
</w:tblStylePr>
```

The insideH element specifies a ¼ point border of type double. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.24 insideV (Table Inside Vertical Edges Border)

This element specifies the border which shall be displayed on all vertical table cell borders which are not on an outmost edge of the parent table (all horizontal borders which are not the leftmost or rightmost border). The appearance of this table cell border in the document shall be determined by the following settings:

- The display of the border on interior edges is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the inside vertical borders of this table shall have the border specified by the associated table style. If no inside vertical edge border is specified in the style hierarchy, then those cells in this table shall not have an inside vertical edge border.

[*Example*: Consider a table in which the table specifies a border on all interior horizontal and vertical edges, as follows:

R1C1	R1C2
R2C1	R2C2

This interior horizontal cell border is specified using the following WordprocessingML:

```
<w:tblPr>
  <w:tblBorders>
    <w:insideH w:val="thickThinSmallGap" w:sz="24" w:space="0"
      w:color="auto"/>
    <w:insideV w:val="thickThinSmallGap" w:sz="24" w:space="0"
      w:color="auto"/>
  </w:tblBorders>
  …
</w:tblPr>
```

The insideV element specifies a 3-point border of type thickThinSmallGap. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.25 insideV (Table Cell Inside Vertical Edges Border)

This element specifies the border which shall be displayed on all interior vertical edges of the current group of table cells. [*Note*: Although individual table cells have no concept of an internal edge, which would render this property useless in most cases, it is used to determine the cell borders to apply to a specific group of cells as part of table conditional formatting in a table style, for example, the inside vertical edges on the set of cells in the header row. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the specified conditional formatting on the table shall not change the current set of internal edge borders on its set of table cells (i.e. their current setting shall remain unchanged).

[*Example*: Consider a table in which the conditional formatting on the header row in the associated table style specifies a double line red cell border for all internal vertical lines as follows:

R1C1	R1C2
R2C1	R2C2

This inner vertical edge cell border is specified using the following WordprocessingML:

```
<w:tblStylePr w:type="firstRow">
  <w:tcPr>
    <w:tcBorders>
      <w:insideV w:val="double" w:sz="4" w:space="0" w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
</w:tblStylePr>
```

The insideV element specifies a ¼ point border of type double. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.26 jc (Table Alignment Exception)

This element specifies the alignment of the set of rows which are part of the current table properties exception list with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this property is used to determine how the table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the bidiVisual element (§17.4.1).

If this property is omitted on a table, then the justification shall be determined by the default set of table properties on the parent table.

[*Example*: Consider the following WordprocessingML table, centered on the text margins with a subset of its rows justified to the left margin by a table property exception:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

That exception would be specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:jc w:val="start"/>
</w:tblPrEx>
```

The jc element specifies that the rows which are part of the table properties exception table must be `left` aligned with respect to the text margins. *end example*]

Attributes	Description

val (Alignment Type)	Specifies the justification which should be applied to the parent table.
	
	[*Example*: Consider the following WordprocessingML fragment for a table in a document:
	
	`<w:tblPr>`
	`  <w:jc w:val="center" />`
	`</w:tblPr>`
	
	This table is now centered on the page. *end example*]
	
	The possible values for this attribute are defined by the ST_JcTable simple type (§17.18.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_JcTable) is located in §A.1. *end note*]

17.4.27 jc (Table Row Alignment)

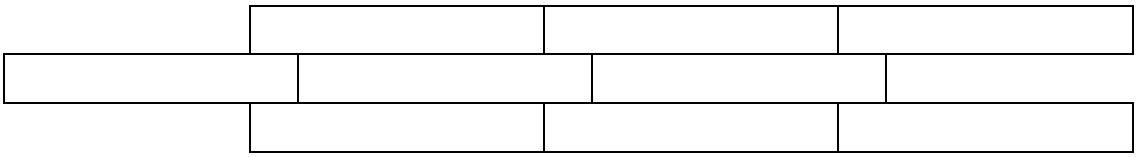
This element specifies the alignment of a single row in the parent table with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this property is used to determine how a specific row in that table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the bidiVisual element (§17.4.1).

If this property is omitted on a table, then the justification shall be determined by the default set of table properties on the parent table.

[*Example*: Consider the following WordprocessingML table, centered on the text margins with its second rows justified to the left margin by a table row level justification:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



That row level setting would be specified using the following WordprocessingML:

```
<w:trPr>
  <w:jc w:val="start"/>
</w:trPr>
```

The jc element specifies that the rows which are part of the table properties exception table must be `left` aligned with respect to the text margins. *end example*]

Attributes	Description

val (Alignment Type)	Specifies the justification which should be applied to the parent table.
	
	[*Example*: Consider the following WordprocessingML fragment for a table in a document:
	
	`<w:tblPr>`
	`  <w:jc w:val="center" />`
	`</w:tblPr>`
	
	This table is now centered on the page. *end example*]
	
	The possible values for this attribute are defined by the ST_JcTable simple type (§17.18.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_JcTable) is located in §A.1. *end note*]

17.4.28 jc (Table Alignment)

This element specifies the alignment of the current table with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this property is used to determine how the table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the bidiVisual element (§17.4.1).

If this property is omitted on a table, then the justification shall be determined by the associated table style. If this property is not specified in the style hierarchy, then the table shall be left justified with zero indentation from the leading margin (the left margin in a left-to-right table or the right margin in a right-to-left table).

[*Example*: Consider the following WordprocessingML table, justified to the left margin by default:

R1C1	R1C2	R1C3

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

R2C1	R2C2	R2C3

This table does not fill the entire width of the text margins. If the table should be right justified to the margin, as follows:

R1C1	R1C2	R1C3
R2C1	R2C2	R2C3

That requirement would be specified using the following WordprocessingML:

```
<w:tblPr>
  <w:jc w:val="end"/>
</w:tblPr>
```

The jc element specifies that the table must be `right` aligned with respect to the text margins. *end example*]

Attributes	Description

val (Alignment Type)	Specifies the justification which should be applied to the parent table.
	
	[*Example*: Consider the following WordprocessingML fragment for a table in a document:
	
	`<w:tblPr>`
	`  <w:jc w:val="center" />`
	`</w:tblPr>`
	
	This table is now centered on the page. *end example*]
	
	The possible values for this attribute are defined by the ST_JcTable simple type (§17.18.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_JcTable) is located in §A.1. *end note*]

17.4.29 noWrap (Don't Wrap Cell Content)

This element specifies how this table cell shall be laid out when the parent table is displayed in a document. This setting only affects the behavior of the cell when the tblLayout for this row (§17.4.52; §17.4.53) is set to use the `auto` algorithm.

This setting shall be interpreted in the context of the tcW element (§17.4.71) as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- If the table cell width has a type attribute value of `fixed`, then this element specifies that that this table cell shall never be smaller than that fixed value when other cells on the line are not at their absolute minimum width.
- If the table cell width has a type attribute value of `pct` or `auto`, then this element specifies that when running the auto fit algorithm, the contents of that this table cell shall be treated as though they have no breaking characters (the contents should be treated as a single contiguous non-breaking string)

If this element is omitted, then cell content shall be allowed to wrap (the cell can be shrunk as needed if it is a fixed preferred width value, and the contents shall be treated as having breaking characters if it is a percentage or automatic width value).

[*Example*: Consider the following three row by three column WordprocessingML table:

In this table, each cell has a `fixed` preferred width of 2.38 inches (3427 twentieths of a point), and the tblLayout for this row (§17.4.52; §17.4.53) is set to use the auto algorithm. If a long non breaking string is added to the middle row, as follows, the two cells are adjusted to override their preferences and accommodate the string:

	ssssss

This element's content model is defined by the common boolean property definition in §17.17.4.

17.4.30 shd (Table Shading Exception)

This element specifies the shading which shall be applied to all cells in the current row as part of a set of table-level property exceptions. Similarly to paragraph shading, this shading shall be applied to the contents of the tab up to the table borders, regardless of the presence of text - unlike cell shading, table shading shall include any cell padding. This property shall be superseded by any cell-level shading on any cell in this row (§17.4.32).

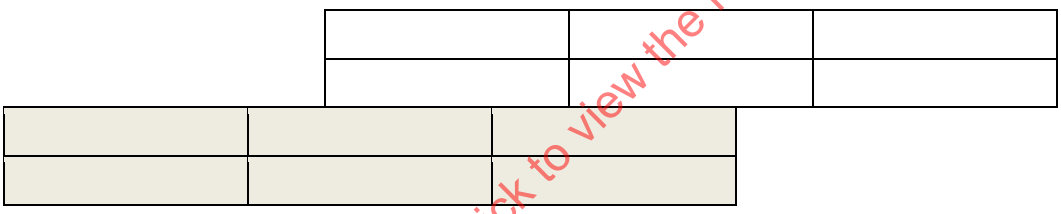
This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cell shading shall be determined by the table-level cell shading settings (§17.4.31) for the current table.

[*Example*: Consider a table in which the final two rows have a set of table-level property exceptions giving them theme color shading in the `background2` theme color, as follows:



This table-level shading exception would be specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:jc w:val="start" />
  <w:shd w:val="clear" w:color="auto" w:fill="EEECE1" w:themeFill="background2"
/>
</w:tblPrEx>
```

The `shd` element specifies cell shading with a `clear` pattern using a background theme color of `background2`. *end example*]

This element's content model is defined by the common shading properties definition in §17.3.5.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.4.31 shd (Table Shading)

This element specifies the shading which shall be applied to the extents the current table. Similarly to paragraph shading, this shading shall be applied to the contents of the tab up to the table borders, regardless of the presence of text - unlike cell shading, table shading shall include any cell padding. This property shall be superseded by any cell-level shading via any table-level property exceptions (§17.4.30); or on any cell in this row (§17.4.32).

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cells within this table shall have the shading specified by the associated table style. If no cell shading is specified in the style hierarchy, then the cells in this table shall not have any cell shading (i.e. they shall be transparent).

[*Example*: Consider a table in which the first cell in the first row has cell-level red shading, as follows:

R1C1	R1C2
R2C1	R2C2

This table level cell shading would be specified using the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:shd w:val="clear" w:color="auto" w:fill="FF0000"/>
    …
  </w:tblPr>
  …
</w:tbl>
```

The shd element specifies cell shading with a `clear` pattern using a background color of `FF0000` (red). *end example*]

This element's content model is defined by the common shading properties definition in §17.3.5.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.4.32 shd (Table Cell Shading)

This element specifies the shading which shall be applied to the extents of the current table cell. Similarly to paragraph shading, this shading shall be applied to the contents of the cell up to the cell borders, regardless of the presence of text.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cell shading shall be determined by the table-level or table-level exception cell shading settings (§17.4.30;§17.4.31) for the current table.

[*Example*: Consider a table in which the first cell in the first row has cell-level red shading, as follows:

R1C1	R1C2
R2C1	R2C2

This cell shading would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:shd w:val="clear" w:color="auto" w:fill="FF0000" />
  </w:tcPr>
</w:tc>
```

The shd element specifies cell shading with a `clear` pattern using a background color of `FF0000` (red). *end example*]

This element's content model is defined by the common shading properties definition in §17.3.5.

17.4.33 start (Table Cell Leading Edge Border)

This element specifies the border which shall be displayed on the leading edge of the current table cell (left for LTR tables, right for RTL tables). The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then the leading edge of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

[*Example*: Consider an LTR table in which the second cell in the first row specifies a leading edge cell border:

R1C1	R1C2
R2C1	R2C2

This leading edge cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
      <w:start w:val="double" w:sz="4" w:space="0" w:color="FF0000" />
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The start element specifies a ½ point border of type `double` on the leading edge of the table cell. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.34 start (Table Cell Leading Margin Default)

This element specifies the amount of space which shall be left between the leading edge of the cell contents and the leading edge of all table cells within the parent table (or table row). This setting can be overridden by the table cell leading margin definition specified by the start element contained within the table cell's properties (§17.4.35).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a leading margin is never specified in the style hierarchy, this table shall have 115 twentieths of a point (0.08 inches) left cell padding by default (excepting individual cell overrides).

[*Example*: Consider a two by two LTR table in which the default table cell leading margin is specified to be exactly 0.25 inches, as follows (marked with an arrow in the first table cell below):

↔ R1C1	R2C1
R2C1	R2C2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This table property is specified using the following WordprocessingML markup:

```
<w:tbl>
  <w:tblPr>
    <w:tblCellMar>
      <w:start w:w="360" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  …
</w:tbl>
```

Every cell in the table has a default leading cell margin setting it to 360 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.35 start (Table Cell Leading Margin Exception)

This element specifies the amount of space which shall be left between the leading extent of the current cell contents and the leading edge border of a specific individual table cell within a table. This setting shall override the table cell leading margin definition specified by the start element contained within the table properties (§17.4.34).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If omitted, then this table cell shall use the leading cell margins defined in the start element contained within the table properties (§17.4.34).

[*Example*: Consider a two row, two column LTR table in which the first table cell in the second row has a leading margin which is specified via an exception to be 0.5 inches, causing the text to be position 0.5" inside the cell, as follows:

R1C1	R1C2
R2C1	R2C2

The exception on this cell would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcMar>
      <w:start w:w="720" w:type="dxa" />
    </w:tcMar>
  </w:tcPr>
</w:tc>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The R2C1 cell in this table has an exception applied to the table cell leading margin setting it to 720 twentieths of a point (0.5 inches). *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.36 start (Table Leading Edge Border)

This element specifies the border which shall be displayed at the leading edge of the current table (left for LTR tables, right for RTL tables). The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the leading edge of this table shall have the border specified by the associated table style. If no leading edge border is specified in the style hierarchy, then this table shall not have a left border.

[*Example*: Consider an LTR table in which the table properties specify a leading edge table border, as follows:

R1C1	R1C2
R2C1	R2C2

This leading edge table border is specified using the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:tblBorders>
      <w:start w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="D0D0D0" w:themeColor="accent3" w:themeTint="99"/>
    </w:tblBorders>
  </w:tblPr>
  …
</w:tbl>
```

The start element specifies a three point leading edge table border of type thinThinThickMediumGap. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.37 tbl (Table)

This element specifies the contents of a table present in the document. A *table* is a set of paragraphs (and other block-level content) arranged in *rows* and *columns*. Tables in WordprocessingML are defined via the tbl element, which is analogous to the HTML table tag.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

When two tbl elements having the same style (§17.4.62) are present within the document content, without any intervening p elements, the corresponding tables shall be treated as a single table.

[*Example*: Consider an empty one-cell table (i.e.; a table with one row, one column) and 1 point borders on all sides:

This table is represented by the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:tblW w:w="5000" w:type="pct"/>
    <w:tblBorders>
      <w:top w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:start w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:bottom w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:end w:val="single" w:sz="4" w:space="0" w:color="auto"/>
    </w:tblBorders>
  </w:tblPr>
  <w:tblGrid>
    <w:gridCol w:w="10296"/>
  </w:tblGrid>
  <w:tr>
    <w:tc>
      <w:tcPr>
        <w:tcW w:w="0" w:type="auto"/>
      </w:tcPr>
      <w:p/>
    </w:tc>
  </w:tr>
</w:tbl>
```

This table specifies table-wide properties of 100% of page width using the tblW element (§17.4.63); a the set of table borders using the tblBorders element (§17.4.38); the table grid which defines a set of shared vertical edges within the table using the tblGrid element (§17.4.48); and a single table row using the tr element (§17.4.78). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tbl) is located in §A.1. *end note*]

17.4.38 tblBorders (Table Borders)

This element specifies the set of borders for the edges of the current table, using the six border types defined by its child elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If the cell spacing for any row is non-zero as specified using the tblCellSpacing element (§17.4.44; §17.4.43; §17.4.45), then there is no border conflict and the table border (or table-level exception border, if one is specified) shall be displayed.

If the cell spacing is zero, then there is a conflict [*Example*: Between the left border of all cells in the first column and the left border of the table. *end example*], which shall be resolved as follows:

- If there is a cell border, then the cell border shall be displayed
- If there is no cell border but there is a table-level exception border on this table row, then the table-level exception border shall be displayed
- If there is no cell or table-level exception border, then the table border shall be displayed

If this element is omitted, then this table shall have the borders specified by the associated table style. If no borders are specified in the style hierarchy, then this table shall not have any table borders.

[*Example*: Consider a table with no associated table style, which defines a set of table borders via direct formatting as follows:

These table borders are specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tbl>
  <w:tblPr>
    <w:tblW w:w="0" w:type="auto"/>
    <w:tblBorders>
      <w:top w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:start w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:bottom w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:end w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:insideH w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:insideV w:val="single" w:sz="4" w:space="0" w:color="000000"
w:themeColor="text1"/>
    </w:tblBorders>
    …
  </w:tblPr>
  …
</w:tbl>
```

The tblBorders element specifies the set of table borders applied to the current table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblBorders) is located in §A.1. *end note*]

17.4.39 tblBorders (Table Borders Exceptions)

This element specifies the set of borders for the edges of the parent table row via a set of table-level property exceptions, using the six border types defined by its child elements.

If the cell spacing for any row is non-zero as specified using the tblCellSpacing element (§17.4.44; §17.4.43; §17.4.45), then there is no border conflict and the table-level exception border shall be displayed.

If the cell spacing is zero, then there is a conflict [*Example*: Between the left border of all cells in the first column and the left border of the table-level exceptions. *end example*], which shall be resolved as follows:

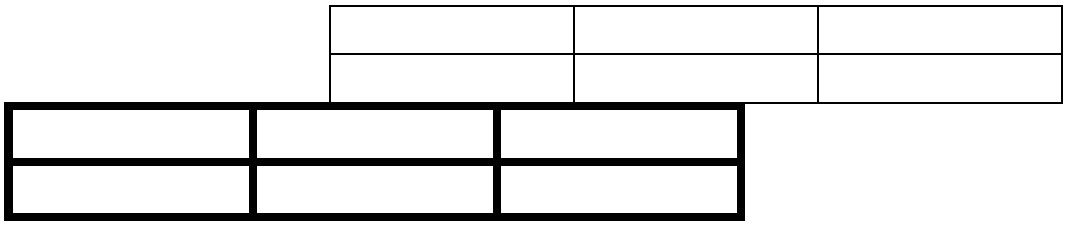
- If there is a cell border, then the cell border shall be displayed
- If there is no cell border, then the table-level exception border shall be displayed

If this element is omitted, then this table shall have the borders specified by the associated table level borders (§17.4.38).

[*Example*: Consider a table in which the final two rows have a set of table-level property exceptions giving them a thicker set of table borders, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



These table borders are specified via a set of table-level property exceptions using the following WordprocessingML:

```
<w:tr>
  <w:tblPrEx>
    <w:tblBorders>
      <w:top w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:start w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:bottom w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:end w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:insideH w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
      <w:insideV w:val="single" w:sz="24" w:space="0" w:color="000000"
w:themeColor="text1"/>
    </w:tblBorders>
  </w:tblPrEx>
</w:tr>
```

The tblBorders element specifies the set of table borders applied to the final two rows in this table as part of the table-level property exceptions. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblBorders) is located in §A.1. *end note*]

17.4.40 tblCaption (Table Caption)

This element specifies the caption for the table.

[*Example*: Consider a table which specifies a caption. This object might contain the following XML markup:

```
<w:tbl>
    <w:tblPr>
        <w:tblCaption w:val="Here is the caption of the table" />
        …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      </w:tblPr>
  </w:tbl>
```

end example]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.4.41 tblCellMar (Table Cell Margin Exceptions)

This element specifies a set of cell margins for all cells in the parent table row via a set of table-level property exceptions. These settings can be overridden by the table cell margin definition specified by the tcMar element contained within the table cell's properties (§17.4.41).

If this element is omitted, then it shall inherit the table cell margins from the table-level cell margins (§17.4.42).

[*Example*: Consider a table whose final two rows are defined to have default cell margins of 0.1 inches for all sides via a table-level property exception, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This set of table cell margin exceptions is specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:tblCellMar>
    <w:top w:w="144" w:type="dxa"/>
    <w:start w:w="144" w:type="dxa"/>
    <w:bottom w:w="144" w:type="dxa"/>
    <w:end w:w="144" w:type="dxa"/>
  </w:tblCellMar>
  …
</w:tblPrEx>
```

The tblCellMar element as a child of tblPrEx specifies the set of default cell margins for all cells in final two rows in current table, in this case, 144 twentieths of a point on all sides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblCellMar) is located in §A.1. *end note*]

17.4.42 tblCellMar (Table Cell Margin Defaults)

This element specifies the default cell margin settings for all cells in the current table. These setting can be overridden by the table cell margin definition specified by the tcMar element contained within the table cell's properties (§17.4.68) or by a set of table-level property exceptions (§17.4.41).

If this element is omitted, then it shall inherit the table cell margins from the associated table style. If table margins are never specified in the style hierarchy, then each margin shall use its default margin size (see child element definitions).

[*Example*: Consider a table defined to have default cells margins of 0.1 inches for all sides, as follows:

R1C1	R1C2
R2C1	R2C2

This set of default table cell margins would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:tblPr>
  <w:tblCellMar>
    <w:top w:w="144" w:type="dxa"/>
    <w:start w:w="144" w:type="dxa"/>
    <w:bottom w:w="144" w:type="dxa"/>
    <w:end w:w="144" w:type="dxa"/>
  </w:tblCellMar>
  …
</w:tblPr>
```

The tblCellMar element as a child of tblPr specifies the set of default cell margins for all cells in the current table, in this case, 144 twentieths of a point. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblCellMar) is located in §A.1. *end note*]

17.4.43 tblCellSpacing (Table Row Cell Spacing)

This element specifies the default table cell spacing (the spacing between adjacent cells and the edges of the table) for all cells in the parent row. If specified, this element specifies the minimum amount of space which shall be left between all cells in the table including the width of the table borders in the calculation. It is important to note that row-level cell spacing shall be added inside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a cell without row-level indentation or cell spacing. Row-level cell spacing shall not increase the width of the overall table.

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

[*Example*: Consider a table whose first cell has a six point wide table border, and a table cell spacing value of 0.01 inches. The resulting table would have 0.01 inches of space between each table cell regardless of the width of the cell border, as follows (notice that no border is covered by any other border):

R1C1	R1C2
R2C1	R2C2

end example]

If this element is omitted, then the cells in this row shall inherit the cell spacing from the associated table level properties.

[*Example*: Consider a table where the second row has a cell spacing of 0.1 inches for all sides specified via the table row properties as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This table row cell spacing is specified using the following WordprocessingML:

```
<w:trPr>
  <w:tblCellSpacing w:w="144" w:type="dxa"/>
  …
</w:trPr>
```

The tblCellSpacing element as a child of trPr specifies the default cell spacing between all cells in the current row, in this case 144 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.44 tblCellSpacing (Table Cell Spacing Exception)

This element specifies a table cell spacing exception for all cells in the parent table row as part of a set of table-level property exceptions. If specified, this element specifies the minimum amount of space which shall be left between all cells in the parent row after including the width of the table borders in the calculation. This setting shall be superseded by the row cell spacing value (§17.4.43). It is important to note that table-level cell spacing shall be added outside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a table cell.

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

[*Example*: Consider a table whose first cell has a six point wide table border, and a table cell spacing value of 0.01 inches. The resulting table would have 0.01 inches of space between each table cell regardless of the width of the cell border, as follows (notice that no border is covered by any other border):

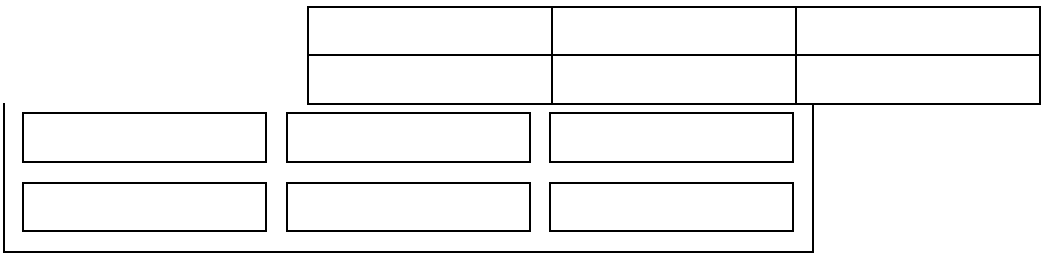
R1C1	R1C2
R2C1	R2C2

end example]

If this element is omitted, then the row shall inherit the table cell spacing from the table-level cell spacing setting (§17.4.42), excepting the case of a row level override.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a table whose final two rows are defined to have cell spacing of 0.1 inches for all sides via a table-level property exception, as follows:



This table cell spacing exception is specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:tblCellSpacing w:w="144" w:type="dxa"/>
  …
</w:tblPrEx>
```

The tblCellSpacing element as a child of tblPrEx specifies the default cell spacing between all cells in final two rows in the current table, in this case 144 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.45 tblCellSpacing (Table Cell Spacing Default)

This element specifies the default table cell spacing (the spacing between adjacent cells and the edges of the table) for all cells in the parent table. If specified, this element specifies the minimum amount of space which shall be left between all cells in the table including the width of the table borders in the calculation. This setting shall be superseded by a table-level exception (§17.4.44) or the row cell spacing value (§17.4.43) in that order. It is important to note that table-level cell spacing shall be added outside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a table cell.

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

[*Example*: Consider a table whose first cell has a six point wide table border, and a table cell spacing value of 0.01 inches. The resulting table would have 0.01 inches of space between each table cell regardless of the width of the cell border, as follows (notice that no border is covered by any other border):

R1C1	R1C2
R2C1	R2C2

end example]

IECNORM.COM: Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then the table shall inherit the table cell spacing from the associated table style. If table cell spacing is never specified in the style hierarchy, no cell spacing shall be added to the parent table.

[*Example*: Consider a table with a default cell spacing of 0.1 inches for all sides as follows:

R1C1	R1C2
R2C1	R2C2

This table cell spacing deafult is specified using the following WordprocessingML:

```
<w:tblPr>
  <w:tblCellSpacing w:w="144" w:type="dxa"/>
  …
</w:tblPr>
```

The tblCellSpacing element as a child of tblPr specifies the default cell spacing between all cells in the current table, in this case 144 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.46 tblDescription (Table Description)

This element specifies the description for the table.

[*Example*: Consider a table which specifies a description. This object might contain the following XML markup:

```
<w:tbl>
    <w:tblPr>
        <w:tblDescription w:val="Here is the description of the table" />
        …
    </w:tblPr>
</w:tbl>
```

end example]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: <w:pPr>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.4.47 tblGrid (Previous Table Grid)

This element specifies a previous table grid state, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table grid settings which were previously in place before a specific set of revisions by one author. The *table grid* is a definition of the set of grid columns which define all of the shared vertical edges of the table, as well as default widths for each of these grid columns. These grid column widths are then used to determine the size of the table based on the table layout algorithm used (§17.4.52;§17.4.53).

[*Example*: Consider the following table with four vertical edges (grid columns):

If we now modify this table by reducing the size of the last column without changing the overall table width, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This table would have a table grid consisting of four grid columns as follows:

```
<w:tblGrid>
  <w:gridCol w:w="2088"/>
  <w:gridCol w:w="1104"/>
  <w:gridCol w:w="3583"/>
  <w:gridCol w:w="2801"/>
  <w:tblGridChange w:id="1">
    <w:tblGrid>
      <w:gridCol w:w="2088"/>
      <w:gridCol w:w="1104"/>
      <w:gridCol w:w="3192"/>
      <w:gridCol w:w="3192"/>
    </w:tblGrid>
  </w:tblGridChange>
</w:tblGrid>
```

The tblGrid element as a child of tblGridChange contains the previous definition for the table grid, consisting of all for grid columns as well as the original widths for those columns. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblGridBase) is located in §A.1. *end note*]

17.4.48 tblGrid (Table Grid)

This element specifies the table grid for the current table. The *table grid* is a definition of the set of grid columns which define all of the shared vertical edges of the table, as well as default widths for each of these grid columns. These grid column widths are then used to determine the size of the table based on the table layout algorithm used (§17.4.52;§17.4.53).

If the table grid is omitted, then a new grid shall be constructed from the actual contents of the table assuming that all grid columns have a width of 0.

[*Example*: Consider the following table with four vertical edges (grid columns):

This table would have a table grid consisting of four grid columns as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tblGrid>
  <w:gridCol w:w="2088"/>
  <w:gridCol w:w="1104"/>
  <w:gridCol w:w="3192"/>
  <w:gridCol w:w="3192"/>
</w:tblGrid>
```

The tblGrid element contains the current definition for the table grid, consisting of all for grid columns as well as default widths for those columns. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblGrid) is located in §A.1. *end note*]

17.4.49 tblHeader (Repeat Table Row on Every New Page)

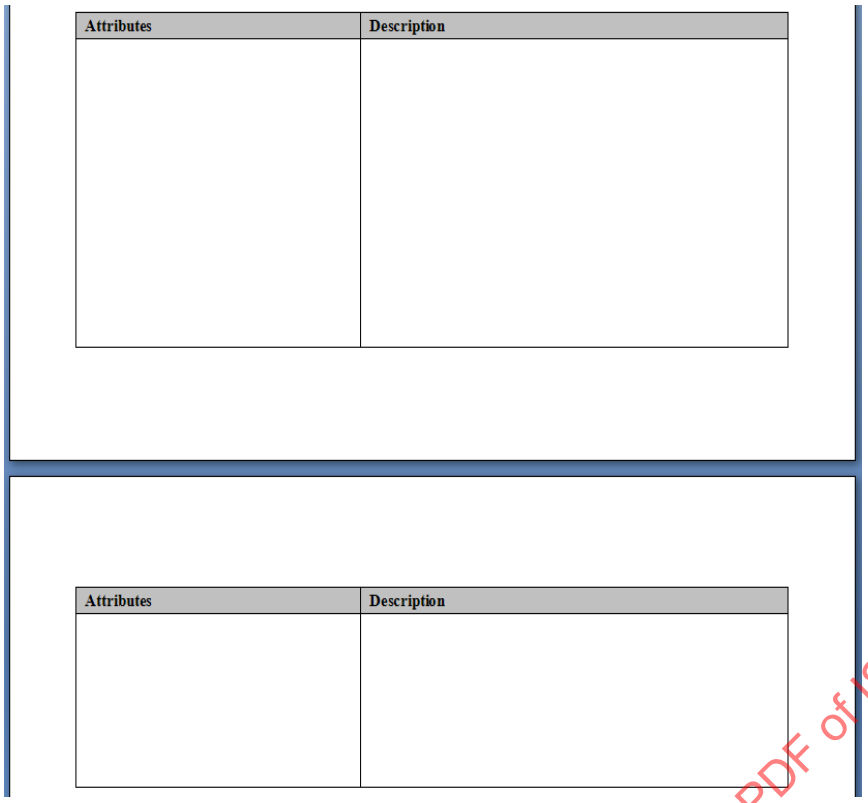
This element specifies that the current table row shall be repeated at the top of each new page on which part of this table is displayed. This gives this table row the behavior of a 'header' row on each of these pages. This element can be applied to any number of rows at the top of the table structure in order to generate multi-row table headers.

If this element is omitted, this table row shall not be repeated on each new page on which the table is displayed. As well, if this row is not contiguously connected with the first row of the table (that is, if this table row is not either the first row, or all rows between this row and the first row are not marked as header rows) then this property shall be ignored.

[*Example*: Consider a table which must have its first row repeated on each new page, like the attribute listings in ISO/IEC 29500, for example:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



Notice that the first row in the table is repeated on the top of the second page. This requirement would be specified as follows in the WordprocessingML for that row:

```
<w:trPr>
  <w:tblHeader />
</w:trPr>
```

The tblHeader element specifies that this table row is repeated as a header row at the top of each page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.4.50 tblInd (Table Indent from Leading Margin)

This element specifies the indentation which shall be added before the leading edge of the current table in the document (the left edge in a left-to-right table, and the right edge in a right-to-left table). This indentation should shift the table into the text margin by the specified amount.

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then the table shall inherit the table indentation from the associated table style. If table indentation is never specified in the style hierarchy, no indentation shall be added to the parent table. If

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

the resulting justification on any table row is not `left` after applying the value of the jc element from the three levels of this property (§17.4.26;§17.4.27;§17.4.28), then this property shall be ignored.

[*Example*: Consider a table which must be indented one inch from the left margin, as follows:

R1C1	R1C2
R2C1	R2C2

This setting would be specified using the following WordprocessingML:

```
<w:tblPr>
  <w:jc w:val="start"/>
  <w:tblInd w:w="1440" w:type="dxa"/>
</w:tblPr>
```

If the properties on this table were now modified to justify it on the right side by setting the value of the jc element to `right`, as follows:

```
<w:tblPr>
  <w:jc w:val="end"/>
  <w:tblInd w:w="1440" w:type="dxa"/>
</w:tblPr>
```

This table would now have no indent, as the justification is no longer on the leading edge (`left`):

R1C1	R1C2
R2C1	R2C2

end example]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.51 tblInd (Table Indent from Leading Margin Exception)

This element specifies the indentation which shall be added before the leading edge of the set of parent table rows which have this set of table-level property exceptions applied. This indentation should shift the table into the text margin by the specified amount in the document (the left edge in a left-to-right table, and the right edge in a right-to-left table).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then the table shall inherit the table indentation from the associated table level property setting. If the resulting justification on the parent table row is not `left` after applying the value of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

jc element from the three levels of this property (§17.4.26;§17.4.27;§17.4.28), then this property shall be ignored.

[*Example*: Consider a table in which the last two rows must be indented one inch from the left margin via a table-level property exception definition, as follows:

This setting would be specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:tblInd w:w="1440" w:type="dxa"/>
</w:tblPrEx>
```

The tblInd element as a child of tblPrEx specifies that the rows with the table-level property exception must be indented by 1440 twentieths of a point (one inch). *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.52 tblLayout (Table Layout)

This element specifies the algorithm which shall be used to lay out the contents of this table within the document. When a table is displayed in a document, it can either be displayed using a fixed width or autofit layout algorithm (each discussed in the simple type referenced by the val attribute).

If this element is omitted, then the value of this element shall be assumed to be auto.

[*Example*: Consider a table which must use the fixed width table layout algorithm. This requirement is specified using the following WordprocessingML:

```
<w:tblPr>
  <w:tblLayout w:type="fixed"/>
</w:tblPr>
```

The tblLayout element specifies that the table must use the fixed layout algorithm. *end example*]

Attributes	Description
type (Table Layout Setting)	Specifies the algorithm which shall be used to lay out the contents of the parent table (see simple type definition for details on each algorithm used). [*Example*: Consider a table which must use the AutoFit width table layout algorithm. This requirement is specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:tblPr> <w:tblLayout w:type="autofit"/> </w:tblPr> The tblLayout element specifies that the table must use the auto layout algorithm. *end example*] The possible values for this attribute are defined by the ST_TblLayoutType simple type (§17.18.87).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblLayoutType) is located in §A.1. *end note*]

17.4.53 tblLayout (Table Layout Exception)

This element specifies the algorithm which shall be used to lay out the contents of all rows with this table within the table which have the set of table-level property exceptions specified by the parent element. When a table is displayed in a document, it can either be displayed using a fixed width or autofit layout algorithm (each discussed in the simple type referenced by the val attribute).

If this element is omitted, then the value of this element shall be assumed to be auto.

[*Example*: Consider a table which must use the fixed width table layout algorithm. This requirement is specified using the following WordprocessingML:

```
<w:tblPrEx>
  <w:tblLayout w:type="fixed"/>
</w:tblPrEx>
```

The tblLayout element specifies that the table must use the fixed layout algorithm. *end example*]

Attributes	Description
type (Table Layout Setting)	Specifies the algorithm which shall be used to lay out the contents of the parent table (see simple type definition for details on each algorithm used). [*Example*: Consider a table which must use the AutoFit width table layout algorithm. This requirement is specified using the following WordprocessingML: <w:tblPr> <w:tblLayout w:type="autofit"/> </w:tblPr> The tblLayout element specifies that the table must use the auto layout algorithm. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TblLayoutType simple type (§17.18.87).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblLayoutType) is located in §A.1. *end note*]

17.4.54 tblLook (Table Style Conditional Formatting Settings Exception)

This element specifies the components of the conditional formatting of the referenced table style (if one exists) which shall be applied to the set of table rows with the current table-level property exceptions. A table style can specify up to six different optional conditional formats [*Example*: Different formatting for first column. *end example*], which then can be applied or omitted from individual table rows in the parent table.

The default setting is to apply the row and column banding formatting, but not the first row, last row, first column, or last column formatting.

[*Example*: Consider a table which must use the following conditional formatting properties from the referenced table style:

- First row conditional formatting
- Last row conditional formatting
- No row banding formatting
- No column banding formatting

The resulting WordprocessingML would be specified as follows:

```
<w:tblPrEx>
  <w:tblLook w:firstRow="true" w:lastRow="true" w:noHBand="true"
w:noVBand="true" />
</w:tblPrEx>
```

The tblLook element specifies which components of the table style are applied to the current table. *end example*]

Attributes	Description
firstColumn (First Column)	Specifies that the first column conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRow (First Row)	Specifies that the first row conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastColumn (Last	Specifies that the last column conditional formatting shall be applied to the table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Column)	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRow (Last Row)	Specifies that the last row conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
noHBand (No Horizontal Banding)	Specifies that the horizontal banding conditional formatting shall not be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
noVBand (No Vertical Banding)	Specifies that the vertical banding conditional formatting shall not be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblLook) is located in §A.1. *end note*]

17.4.55 tblLook (Table Style Conditional Formatting Settings)

This element specifies the components of the conditional formatting of the referenced table style (if one exists) which shall be applied to the current table. A table style can specify up to six different optional conditional formats [*Example*: Different formatting for first column. *end example*], which then can be applied or omitted from individual tables in the document.

The default setting is to apply the row and column banding formatting, but not the first row, last row, first column, or last column formatting.

[*Example*: Consider a table which must use the following conditional formatting properties from the referenced table style:

- First row conditional formatting
- Last row conditional formatting
- No row banding formatting
- No column banding formatting

The resulting WordprocessingML would be specified as follows:

```
<w:tblPr>
  <w:tblLook  w:firstRow="true" w:lastRow="true" w:noHBand="true"
w:noVBand="true" />
</w:tblPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The tblLook element specifies which components of the table style are applied to the current table. *end example*]

Attributes	Description
firstColumn (First Column)	Specifies that the first column conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
firstRow (First Row)	Specifies that the first row conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastColumn (Last Column)	Specifies that the last column conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
lastRow (Last Row)	Specifies that the last row conditional formatting shall be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
noHBand (No Horizontal Banding)	Specifies that the horizontal banding conditional formatting shall not be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
noVBand (No Vertical Banding)	Specifies that the vertical banding conditional formatting shall not be applied to the table. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblLook) is located in §A.1. *end note*]

17.4.56 tblOverlap (Floating Table Allows Other Tables to Overlap)

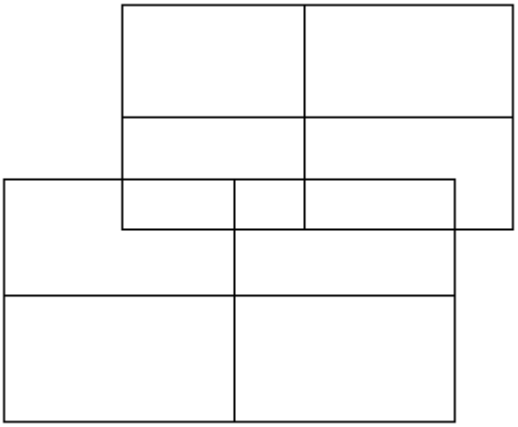
This element specifies whether the current table shall allow other floating tables to overlap its extents when the tables are displayed in a document. If specified, then no adjustment shall be made to prevent tables whose properties would normally cause them to overlap from overlapping when displayed. If turned off, then the tables shall be adjusted as needed to prevent them from overlapping when displayed by adjusting the floating table properties as needed.

If this element is omitted on a given table, then this table shall allow other tables to overlap when displayed. If the parent table is not floating via the tblpPr element (§17.4.57), then this element shall be ignored.

[*Example*: Consider two floating tables in a WordprocessingML document which overlap when displayed, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

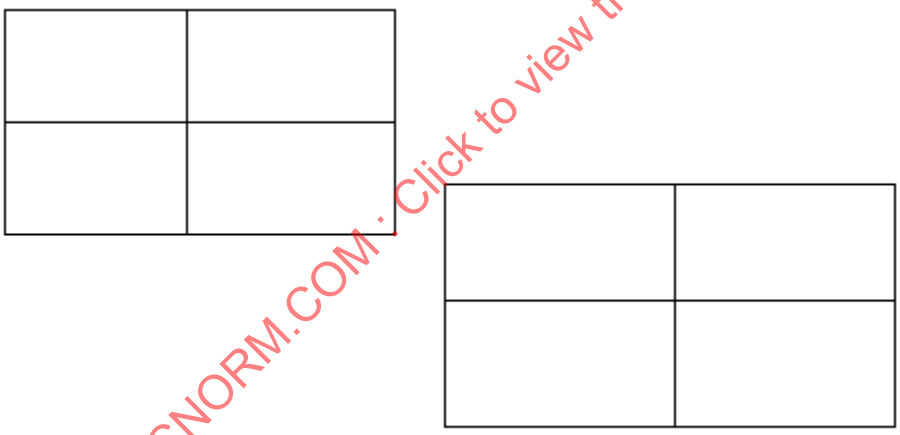
©ISO/IEC 2016 – All rights reserved



If either of these tables specifies that it must not allow overlapping, using the following WordprocessingML:

```
<w:tblPr>
  <w:tblOverlap w:val="never"/>
</w:tblPr>
```

The resulting tables must not overlap, and must be adjusted at display time to prevent any overlapping, for example:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The tblOverlap element with a value of never specifies that the specified table cannot overlap with other floating tables in the document. *end example*]

Attributes	Description
val (Floating Table Overlap Setting)	Specifies whether a floating table shall allow other floating tables in the document to overlap its extents when displayed.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: The following WordprocessingML specifies that the table is not allowed to overlap: `<w:tblOverlap w:val="never" />` *end example*] The possible values for this attribute are defined by the ST_TblOverlap simple type (§17.18.88).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblOverlap) is located in §A.1. *end note*]

17.4.57 tblpPr (Floating Table Positioning)

This element specifies information about the current table with regard to floating tables. *Floating tables* are tables in a document which are not part of the main text flow in the document, and are instead absolutely positioned with a specific size and position relative to non-frame content in the current document.

The first piece of information specified by the tblpPr element is that the current table is actually a floating table. This information is specified simply by the presence of the tblpPr element in table's properties. If the tblpPr element is omitted, the table shall not floating in the document.

The second piece of information is the positioning of the table, which is specified by the attribute values stored on the tblpPr element. In all absolute positioning cases, the positioning of the table is relative to its top-left corner position. For relative positioning (e.g. center), the positioning of the table is relative to its entire frame.

Note that the table still has a logical position in the file (its location within the block-level elements in the document). This logical location shall be used to calculate the position of the table relative to a paragraph, using the next regular (non-table, non-frame) paragraph in the document.

[*Example*: Consider a floating table which is positioned three inches from the edge of the page extents on both its top and left edges (i.e. the top-left corner occurs at 3" x 3"). This floating table would be specified using the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:tblpPr w:leftFromText="144" w:rightFromText="144" w:topFromText="144"
w:bottomFromText="144" w:vertAnchor="page" w:horzAnchor="page" w:tblpX="4320"
w:tblpY="4320"/>
    …
  </w:tblPr>
  …</w:tbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The presence of the tblpPr element dictates that this table is a floating table, and its attributes specify that the floating table shall be anchored 4320 twentieths of a point (3 inches) from the top and left edges of the current page. *end example*]

Attributes	Description

bottomFromText (Distance From Bottom of Table to Text)	Specifies the minimum distance which shall be maintained between the current floating table and the top of text in the paragraph which is below this floating table.
	
	This distance is expressed in twentieths of a point.
	
	If this attribute is omitted, its value shall be assumed to be 0.
	
	[*Example*: Consider a floating table which should have a minimum of a one-half inch spacing from any text on its bottom side. This constraint would be specified using the following WordprocessingML:
	
	`<w:tblPr>`
	`  <w:tblpPr … w:bottomFromText="720" />`
	`</w:tblPr>`
	
	The bottomFromText attribute specifies that the spacing between text and this floating table shall be a minimum of 720 twentieths of a point. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

horzAnchor (Table Horizontal Anchor)	Specifies the base object from which the horizontal positioning in the tblpX and/or tblpXSpec attribute should be calculated.
	
	A floating table can be horizontally positioned relative to:
	• The vertical edge of the page before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs)
	• The vertical edge of the text margin before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs)
	• The vertical edge of the text margin for the column in which the anchor paragraph is located
	
	If this attribute is omitted, then its value shall be assumed to be page.
	
	[*Example*: Consider a floating table which should be positioned one inch to the right of its column in a left-to-right document. This floating table would be specified using the following WordprocessingML:
	
	`<w:tblPr>`
	`  <w:tblpPr … w:tblpX="1440" w:horzAnchor="margin" />`
	`</w:tblPr>`
	
	These table properties specify that they are relative to the current margin, and that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	relative to that column, the floating table should be 1440 twentieths of a point. *end example*] The possible values for this attribute are defined by the ST_HAnchor simple type (§17.18.35).

leftFromText (Distance From Left of Table to Text)	Specifies the minimum distance which shall be maintained between the current floating table and the edge of text in the paragraph which is to the left of this floating table.
	
	This distance is expressed in twentieths of a point.
	
	If this attribute is omitted, its value shall be assumed to be 0.
	
	[*Example*: Consider a floating table which should have a minimum of a one-half inch spacing from any text on its left. This constraint would be specified using the following WordprocessingML:
	
	`<w:tblPr>`
	`  <w:tblpPr … w:leftFromText="720" />`
	`</w:tblPr>`
	
	The leftFromText attribute specifies that the spacing between text and this floating table must be a minimum of 720 twentieths of a point. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

rightFromText ((Distance From Right of Table to Text)	Specifies the minimum distance which shall be maintained between the current floating table and the edge of text in the paragraph which is to the right of this floating table.
	
	This distance is expressed in twentieths of a point.
	
	If this attribute is omitted, its value shall be assumed to be 0.
	
	[*Example*: Consider a floating table which should have a minimum of a one-half inch spacing from any text on its right. This constraint would be specified using the following WordprocessingML:
	
	`<w:tblPr>`
	`  <w:tblpPr … w:rightFromText="720" />`
	`</w:tblPr>`
	
	The rightFromText attribute specifies that the spacing between text and this floating table must be a minimum of 720 twentieths of a point. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

tblpX (Absolute Horizontal Distance From Anchor)	Specifies an absolute horizontal position for the floating table. This absolute position is specified relative to the horizontal anchor specified by the horzAnchor attribute for this floating table.
	
	This value is expressed in twentieths of a point. If it is positive, then the floating table is positioned after the anchor object in the direction of horizontal text flow in this document. If it is negative, then the floating table is positioned before the anchor object in the direction of horizontal text flow in this document.
	
	If the tblpXSpec attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a floating table:
	
	`<w:tbl>`
	`  <w:tblPr>`
	`  <w:tblpPr … w:horzAnchor="page" w:tblpX="1643"/>`
	`  </w:tblPr>`
	`  …`
	`</w:tbl>`
	
	This floating table specifies that it should be located exactly 1643 twentieths of a point after the vertical edge of the page (from the horizAnchor attribute). *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

tblpXSpec (Relative Horizontal Alignment From Anchor)	Specifies a relative horizontal position for the floating table. This relative position is specified relative to the horizontal anchor specified by the horizAnchor attribute for this floating table.
	
	If omitted, this attribute is not specified and the value of the tblpX attribute determines the absolute horizontal position of the floating table. If specified, the position for this attribute supersedes any value which is specified in the tblpX attribute, and that value is ignored.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a floating table:
	
	`<w:tbl>`
	`  <w:tblPr>`
	`  <w:tblpPr … w:horzAnchor="page" w:tblpX="1643"`
	`w:tblpXSpec="left"/>`
	`  </w:tblPr>`
	`  …`
	`</w:tbl>`
	
	This floating table specifies that it has a horizontal placement of exactly 1643 twentieths

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	of a point relative to the page, but that exact placement is overridden by the presence of the tblpXSpec attribute to place the table on the `left` side of the page. *end example*] The possible values for this attribute are defined by the ST_XAlign simple type (§22.9.2.18).

tblpY (Absolute Vertical Distance From Anchor)	Specifies an absolute vertical position for the floating table. This absolute position is specified relative to the vertical anchor specified by the vertAnchor attribute for this floating table.
	
	This value is expressed in twentieths of a point. If it is positive, then the floating table is positioned after the anchor object in the direction of vertical text flow in this document. If it is negative, then the floating table is positioned before the anchor object in the direction of vertical text flow in this document.
	
	If the tblpYSpec attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be `0`.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a floating table:
	
	`<w:tbl>`
	`  <w:tblPr>`
	`  <w:tblpPr … w:vertAnchor="text" w:tblpY="73" />`
	`  </w:tblPr>`
	`  …`
	`</w:tbl>`
	
	This floating table specifies that it should be located exactly 79 twentieths of a point below the top vertical edge of the anchor's paragraph's `text` (from the vertAnchor attribute), assuming that the vertical text direction is top to bottom. *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

tblpYSpec (Relative Vertical Alignment from Anchor)	Specifies a relative vertical position for the floating table. This relative position is specified relative to the vertical anchor specified by the vertAnchor attribute for this floating table.
	
	If omitted, this attribute is not specified and the value of the tblpY attribute determines the absolute horizontal position of the floating table. If specified, the position for this attribute supersedes any value which is specified in the tblpY attribute, and that value is ignored, unless the vertAnchor is set to `text`, in which case any relative positioning is not allowed, and is itself ignored.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a floating table:
	
	`<w:tbl>`
	`  <w:tblPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:tblpPr … w:vertAnchor="margin" w:tblpY="73" w:tblpYSpec="center"/> </w:tblPr> … </w:tbl> This floating table specifies that it has a vertical placement of exactly 73 twentieths of a point relative to the top margin, but that exact placement is overridden by the presence of the tblpYSpec attribute to place the table in the center of the margin. *end example*] The possible values for this attribute are defined by the ST_YAlign simple type (§22.9.2.20).
topFromText (Distance From Top of Table to Text)	Specifies the minimum distance which shall be maintained between the current floating table and the bottom edge of text in the paragraph which is above this floating table. This distance is expressed in twentieths of a point. If this attribute is omitted, its value shall be assumed to be 0. [*Example*: Consider a floating table which should have a minimum of a one-half inch spacing from any text above it. This constraint would be specified using the following WordprocessingML: <w:tblPr> <w:tblpPr … w:topFromText="720" /> </w:tblPr> The topFromText attribute specifies that the spacing between text and this floating table must be a minimum of 720 twentieths of a point. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
vertAnchor (Table Vertical Anchor)	Specifies the base object from which the vertical positioning in the tblpY attribute should be calculated. A floating table can be horizontally positioned relative to: • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the text margin before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) If this attribute is omitted, then its value shall be assumed to be page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a floating table which should be positioned two inches below the page top in a top-to-bottom document. This floating table would be specified using the following WordprocessingML:
	
	`<w:tblPr>`
	`  <w:tblpPr … w:tblpY="2880" w:vertAnchor="page" />`
	`</w:tblPr>`
	
	These floating table properties specify that they are relative to the anchor page, and that relative to that page, the table should be `2880` twentieths of a point in the direction of the flow of text (down, in this case). *end example*]
	
	The possible values for this attribute are defined by the ST_VAnchor simple type (§17.18.100).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPPr) is located in §A.1. *end note*]

17.4.58 tblPr (Previous Table Properties)

This element specifies a previous set of table properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table property settings which were previously in place before a specific set of revisions by one author. These properties affect the appearance of all rows and cells within the parent table, but can be overridden by individual table-level exception, row, and cell level properties, as defined by each property.

[*Example*: Consider the following simple WordprocessingML table:

If the table justification is set to `center` and the table shading to set to red with revision marking on, as follows:

The revision tracked on this table would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tblPr>
  <w:tblStyle w:val="TableGrid"/>
  <w:tblW w:w="0" w:type="auto"/>
  <w:jc w:val="center"/>
  <w:shd w:val="clear" w:color="auto" w:fill="FF0000"/>
  <w:tblLook w:firstRow="true" w:firstColumn="true"
    w:noVBand="true" />
  <w:tblPrChange w:id="0" … >
    <w:tblPr>
      <w:tblStyle w:val="TableGrid"/>
      <w:tblW w:w="0" w:type="auto"/>
      <w:tblLook w:firstRow="true" w:firstColumn="true"
      w:noVBand="true" />
    </w:tblPr>
  </w:tblPrChange>
</w:tblPr>
```

The tblPr element as a child of tblPrChange contains the previous definition for the table properties, consisting of the properties set before the current tracked revision. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrBase) is located in §A.1. *end note*]

17.4.59 tblPr (Table Properties)

This element specifies the set of table-wide properties applied to the current table. These properties affect the appearance of all rows and cells within the parent table, but can be overridden by individual table-level exception, row, and cell level properties as defined by each property.

[*Example*: Consider the following simple WordprocessingML table:

This table defines a one point single border for all border types and is set to 100% of page width - both table-wide properties. The resulting table is represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tblPr>
    <w:tblW w:w="0" w:type="auto"/>
    <w:tblBorders>
      <w:top w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:start w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:bottom w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:end w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:insideH w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:insideV w:val="single" w:sz="4" w:space="0" w:color="auto"/>
    </w:tblBorders>
  </w:tblPr>
  …
</w:tbl>
```

In this example, the tblW element (§17.4.63) defines the total width of the table, which, in this case, is set to a type of auto, which specifies that the table should be automatically sized to fit its contents. The tblBorders element (§17.4.38) specifies each of the table's borders, and specifies a one point border on the top, left, bottom, right and inside horizontal and vertical border. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPr) is located in §A.1. *end note*]

17.4.60 tblPrEx (Table-Level Property Exceptions)

This element specifies a set of table properties which shall be applied to the contents of this row in place of the table properties specified in the tblPr element.

[*Note*: These properties are typically used in cases involving legacy documents, as well as cases where two existing independent tables are merged (in order to prevent the look of the second table from being superseded by the first table). *end note*]

[*Example*: Consider the following two tables in a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

These two tables each have a different set of table level borders. If the interceding paragraphs between these two tables is removed and the tables are merged together, it is obviously undesirable to have the second table lose its formatting and match the properties of the first table. Therefore, when the tables are merged as follows (note that there is now only one table):

The resulting WordprocessingML for the last three rows of the table would include the following set of table-level property exceptions:

```
<w:tr>
  <w:tblPrEx>
    <w:tblBorders>
      <w:top w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
      <w:start w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
      <w:bottom w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
      <w:end w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
      <w:insideH w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
      <w:insideV w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="auto"/>
    </w:tblBorders>
  </w:tblPrEx>
  …
</w:tr>
```

The tblPrEx element contains all table-level properties which are being overridden for the current row in the table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrEx) is located in §A.1. *end note*]

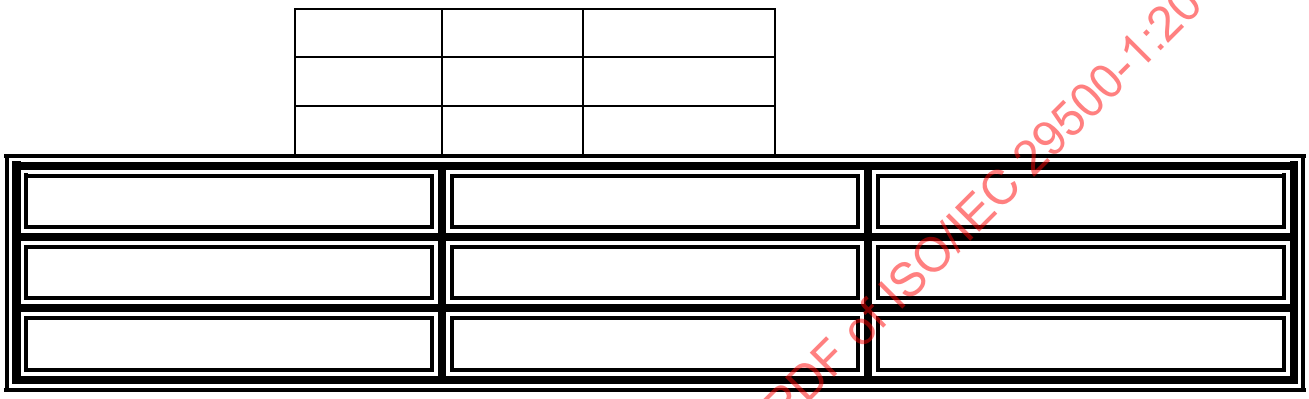
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

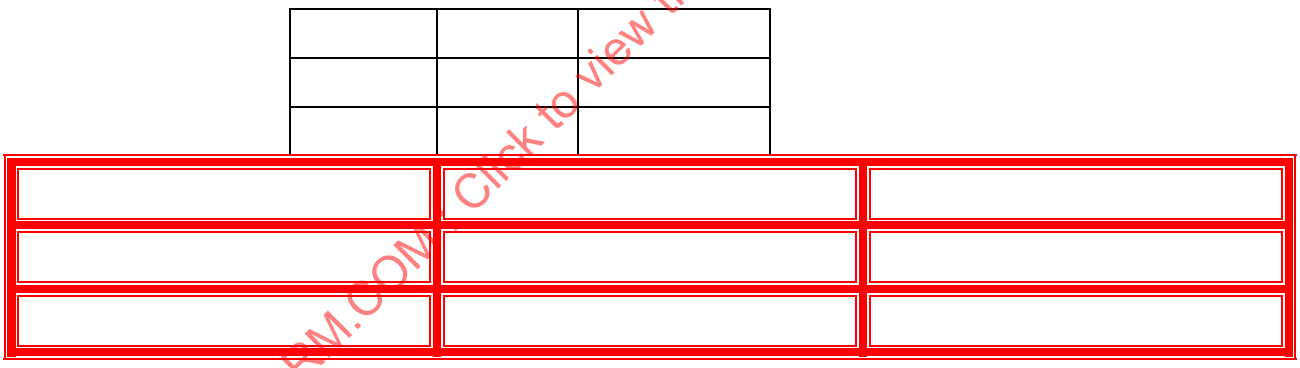
17.4.61 tblPrEx (Previous Table-Level Property Exceptions)

This element specifies a previous set of table-level property exceptions, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table-level property exceptions which were previously in place before a specific set of revisions by one author.

[*Example*: Consider the following two tables in a WordprocessingML document. If the interceding paragraphs between these two tables is removed and the tables are merged together, it is obviously undesirable to have the second table lose its formatting and match the properties of the first table. Therefore, when the tables are merged as follows (note that there is now only one table):



If the border type is changed to a red border of type `thinThickThinSmallGap` with revisions tracked, as follows:



The resulting WordprocessingML for the last three rows of the table would include the following set of table-level property exceptions with revision tracking:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tr>
  <w:tblPrEx>
    <w:tblBorders>
      <w:top w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
      <w:start w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
      <w:bottom w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
      <w:end w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
      <w:insideH w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
      <w:insideV w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
       w:color="auto"/>
    </w:tblBorders>
    <w:tblPrExChange w:id="9" … >
      <w:tblPrEx>
        <w:tblBorders>
          <w:top w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
          <w:start w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
          <w:bottom w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
          <w:end w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
          <w:insideH w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
          <w:insideV w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
           w:color="FF0000"/>
        </w:tblBorders>
      </w:tblPrEx>
    </w:tblPrExChange>
  </w:tblPrEx>
  …
</w:tr>
```

The tblPrEx element as a child of tblPrExChange contains the previous definition for the table-level property exceptions, consisting of the properties set before the current tracked revision. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrExBase) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.4.62 tblStyle (Referenced Table Style)

This element specifies the style ID of the table style which shall be used to format the contents of this table.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles (this element)
- Numbering styles
- Paragraph styles
- Character styles
- Direct Formatting

This means that all properties specified in the style element (§17.7.4.17) with a styleId which corresponds to the value in this element's val attribute are applied to the table at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no table style shall be applied to the current table. As well, this property is ignored if the table properties are themselves part of a table style.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tblPr>
  <w:tblStyle w:val="TestTableStyle" />
</w:tblPr>
```

This table specifies that it inherits all of the table properties specified by the table style with a styleId of `TestTableStyle`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.4.63 tblW (Preferred Table Width)

This element specifies the preferred width for this table. This preferred width is used as part of the table layout algorithm specified by the tblLayout element (§17.4.52; §17.4.53) - full description of the algorithm in the ST_TblLayout simple type (§17.18.87).

All widths in a table are considered preferred because:

- The table shall satisfy the shared columns as specified by the tblGrid element (§17.4.48)
- Two or more widths can have conflicting values for the width of the same grid column
- The table layout algorithm (§17.18.87) can require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type pct for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type auto.

[*Example*: Consider a WordprocessingML table defined as follows:

```
<w:tbl>
  <w:tblPr>
    <w:tblW w:type="dxa" w:w="1440"/>
  </w:tblPr>
  …
</w:tbl>
```

This table specifies that it has a preferred table width of 1440 twentieths of a point (one inch). The resulting table would therefore be sized such that the table maintains that preferred width, as follows:

	Hello world	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The text `Hello world` makes the middle cell larger, and the other two cells are size to maintain the preferred widths of one inch for the overall table width:

	Hello world this is a longer string.	

However, when the middle table cell requires a larger width to accommodate non-breaking text, that preference can be overridden as needed:

	Hello worlddddddddddddddddddddddddddddddd	

In this case, the middle cell's long non breaking string caused the table to be expanded to prevent breaking the string, and therefore to override the preferred width on the table. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.64 tblW (Preferred Table Width Exception)

This element specifies the preferred width for the parent table row via a set of table-level property exceptions. This preferred width is used as part of the table layout algorithm specified by the tblLayout element (§17.4.52n; §17.4.53) - full description of the algorithm in the ST_TblLayout simple type (§17.18.87).

All widths in a table are considered preferred because:

- The table shall satisfy the shared columns as specified by the tblGrid element (§17.4.48)
- Two or more widths can have conflicting values for the width of the same grid column
- The table layout algorithm (§17.18.87) can require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type `pct` for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type `auto`.

[*Example*: Consider a row in a WordprocessingML table defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tr>
  <w:tblPrEx>
    <w:tblW w:type="auto" w:w="1440"/>
  </w:tblPrEx>
  <w:trPr>

  </w:trPr>
  …
</w:tr>
```

This table-level property exception specifies that it has a preferred table width of `1440` twentieths of a point (one inch). The resulting table row would therefore be sized such that the table maintains that preferred width, as follows:

	Hello world	

The text `Hello world` makes the middle cell larger, and the other two cells are size to maintain the preferred widths of one inch for the overall table width:

	Hello world this is a longer string.	

However, when the middle table cell requires a larger width to accommodate non-breaking text, that preference can be overridden as needed:

	Hello worlddddddddddddddddddddddddddddddd	

In this case, the middle cell's long non breaking string caused the table to be expanded to prevent breaking the string, and therefore to override the preferred width on the table row. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.65 tc (Table Cell)

This element specifies a single cell in a table row, which contains the table's content. Table cells in WordprocessingML are analogous to HTML `td` elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

A tc element has one formatting child element, tcPr (§17.4.69), which defines the properties for the cell. Each unique property on the table cell is specified by a child element of this element. As well, a table cell can contain any block-level content, which allows for the nesting of paragraphs and tables within table cells.

If a table cell does not include at least one block-level element, then this document shall be considered corrupt.

[*Example*: Consider a table consisting of a single table cell, which contains the text `Hello, world`:

Hello, world

This table cell's content is represented by the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcW w:w="0" w:type="auto"/>
  </w:tcPr>
  <w:p>
    <w:r>
      <w:t>Hello, world</w:t>
    </w:r>
  </w:p>
</w:tc>
```

The tc element contains a set of cell-level properties defined using the tcPr element, and a single block-level element - in this case, a paragraph. *end example*]

Attributes	Description

id (Table Cell Identifier)	Specifies a unique identifier for the current table cell. This identifier shall be unique within the table, and is used to identify this table cell as a header cell for other cells within the table, using the headers child element.
	
	If this attribute is omitted, this table cell has no unique identifier.
	
	[*Example*: Consider a table cell defined as follows:
	
	`<w:tc w:id="januaryeight">`
	`…`
	`</w:tc>`
	
	The value in the id specifies a unique identifer of `januaryeight`. Other cells in the table are then able to reference this cell as a row or column header by referencing this ID. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tc) is located in §A.1. *end note*]

17.4.66 tcBorders (Table Cell Borders)

This element specifies the set of borders for the edges of the current table cell, using the eight border types defined by its child elements.

If the cell spacing for any row is non-zero as specified using the tblCellSpacing element (§17.4.44; §17.4.43; §17.4.45), then there is never a border conflict (as the non-zero cell spacing is applied above and beyond each individual cell border's width) and all table, table-level exception, and table cell borders shall be displayed.

If the cell spacing is zero, then there can be a conflict between two adjacent cell borders [*Example*: Between the left border of all cells in the second column and the right border of all cells in the first column of the table. *end example*], which shall be resolved as follows:

5. If either conflicting table cell border is `nil` or `none` (no border), then the opposing border shall be displayed.
6. If a cell border conflicts with a table border, the cell border always wins.
7. Each border shall then be assigned a weight using the following formula, and the border value using this calculation shall be displayed over the alternative border:

$$W_{\text{border}} = \#\text{ of lines in border} * \text{border number}$$

1. The border number shall be determined by this list:

single	1
thick	2
double	3
dotted	4
dashed	5
dotDash	6
dotDotDash	7
triple	8
thinThickSmallGap	9
thickThinSmallGap	10
thinThickThinSmallGap	11
thinThickMediumGap	12
thickThinMediumGap	13
thinThickThinMediumGap	14
thinThickLargeGap	15
thickThinLargeGap	16

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

thinThickThinLargeGap	17
wave	18
doubleWave	19
dashSmallGap	20
dashDotStroked	21
threeDEmboss	22
threeDEngrave	23
outset	24
inset	25

2. If the borders have an equal weight, than the higher of the two on this precedence list shall win:

   - single
   - thick
   - double
   - dotted
   - dashed
   - dotDash
   - dotDotDash
   - triple
   - thinThickSmallGap
   - thickThinSmallGap
   - thinThickThinSmallGap
   - thinThickMediumGap
   - thickThinMediumGap
   - thinThickThinMediumGap
   - thinThickLargeGap
   - thickThinLargeGap
   - thinThickThinLargeGap
   - wave
   - doubleWave
   - dashSmallGap
   - dashDotStroked
   - threeDEmboss
   - threeDEngrave
   - outset
   - inset

3. If the borders have an identical style, than each border color shall be assigned a brightness value as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

$$\text{Brightness} = \text{R} + \text{B} + 2 * \text{G}$$

The color with the smaller brightness value shall win.

4. If the borders have an identical brightness value above, than each border color shall be assigned a new brightness value as follows:

$$\text{Brightness} = \text{B} + 2 * \text{G}$$

The color with the smaller brightness value shall win.

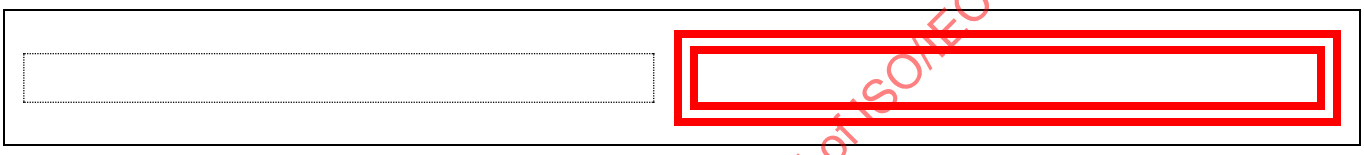
5. If the borders have an identical brightness value above, than each border color shall be assigned a brightness value as follows:

$$\text{Brightness} = \text{G}$$

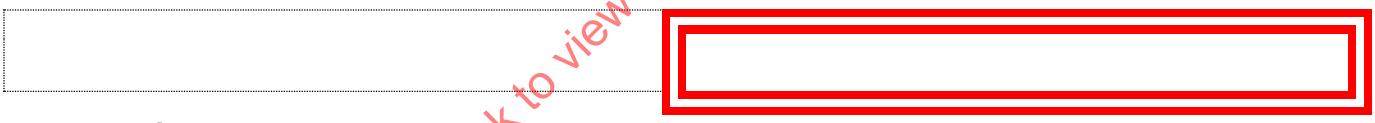
The color with the smaller brightness value shall win.

6. If the borders have an identical brightness value above, then they are functionally identical, and the first border in reading order should be displayed.

[*Example*: Consider the following two cell table (with exaggerated table cell spacing for clarity):



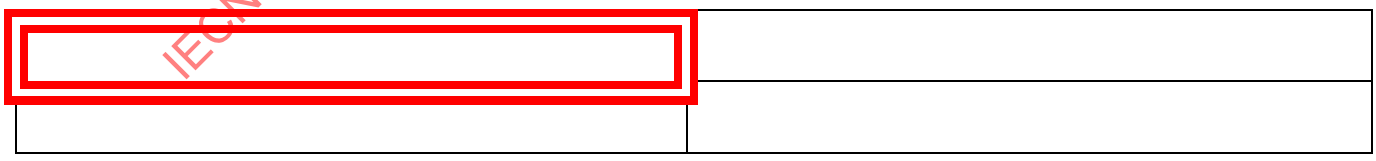
If we collapse the cell spacing, there are conflicting borders at all edges. For each cell/table border conflict, rule #2 says that the cell border must win. For the conflict in the center between two cell borders, rule #3 gives us a larger border weight for the right cell's border, resulting in the following table:



end example]

If this element is omitted, then this table shall have the borders specified by the associated table style. If no borders are specified in the style hierarchy, then this table shall not have any table borders.

[*Example*: Consider a table whose first cell specifies cell-level borders consisting of a think double red line, as follows:



These cell borders are specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:tcPr>
  <w:tcBorders>
    <w:top w:val="double" w:sz="24" w:space="0" w:color="FF0000"/>
    <w:start w:val="double" w:sz="24" w:space="0" w:color="FF0000"/>
    <w:bottom w:val="double" w:sz="24" w:space="0" w:color="FF0000"/>
    <w:end w:val="double" w:sz="24" w:space="0" w:color="FF0000"/>
  </w:tcBorders>
</w:tcPr>
```

The tcBorders element specifies the set of borders applied to the first cell as a 3 point double border. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcBorders) is located in §A.1. *end note*]

17.4.67 tcFitText (Fit Text Within Cell)

This element specifies that the contents of the current cell shall have their inter-character spacing increased or reduced as necessary to fit the width of the text extents of the current cell. This setting shall behave identically to placing the contents of this paragraph in a run and using the fitText element (§17.3.2.14), if the width provided on that element matched the width of the current cell.

If this element is omitted, then the text in this cell shall not be fit to the current cell extents.

[*Example*: Consider a 2 row by two column table, in which the contents of the two cells in the first row have both have the fit text property set, as follows:

```
<w:tcPr>
  <w:tcFitText w:val="true"/>
</w:tcPr>
```

The resulting table cells must have their contents fit to the extents of the parent table cell, as follows:

Sample text in R1C1.	And this table cell instead contains a very very long string of sample text in R2C2.
R2C1	R2C2

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.4.68 tcMar (Single Table Cell Margins)

This element specifies a set of cell margins for a single table cell in the parent table.

This setting, if present, shall override the table cell margins from the table-level cell margins (§17.4.42).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a table whose first cell is defined to have default cell margins of 0.5 inches for all sides rather then the table defaults, as follows:

R1C1
R2C1

This set of table cell margins is specified using the following WordprocessingML:

```
<w:tcPr>
  <w:tcMar>
    <w:top w:w="720" w:type="dxa"/>
    <w:start w:w="720" w:type="dxa"/>
    <w:bottom w:w="720" w:type="dxa"/>
    <w:end w:w="720" w:type="dxa"/>
  </w:tcMar>
  …
</w:tcPr>
```

The tcMar element as a child of tcPr specifies the set of table cell margins used for the first table cell, in this case, 720 twentieths of a point on all sides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcMar) is located in §A.1. *end note*]

17.4.69 tcPr (Table Cell Properties)

This element specifies the set of properties which shall be applied a specific table cell. Each unique property is specified by a child element of this element. In any instance where there is a conflict between the table level, table-level exception, or row level properties with a corresponding table cell property, these properties shall overwrite the table or row wide properties.

[*Example*: Consider a table where the cell width overwrites the table width represented in the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tblPr>
    <w:tblCellMar>
      <w:start w:w="0" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  …
  <w:tr>
    <w:tc>
      <w:tcPr>
        <w:tcMar>
          <w:start w:w="720" w:type="dxa"/>
        </w:tcMar>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
</w:tbl>
```

This table cell has a left cell margin of 720 twentieths of a point (one half inch) as specified in the tcMar element, which overwrites the table level setting of 0 left table cell margin. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcPr) is located in §A.1. *end note*]

17.4.70 tcPr (Previous Table Cell Properties)

This element specifies a previous set of table cell properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table cell property settings which were previously in place before a specific set of revisions by one author. Each unique property is specified by a child element of this element. In any instance where there is a conflict between the table level, table-level exception, or row level properties with a corresponding table cell property, these properties shall overwrite the table or row wide properties.

[*Example*: Consider a basic two row by two column table as follows:

If the cell shading in the first cell is set to red with revision tracking enabled, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This revision is specified as follows in the associated WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcW w:w="4788" w:type="dxa"/>
    <w:shd w:val="clear" w:color="auto" w:fill="FF0000"/>
    <w:tcPrChange w:id="2" …>
      <w:tcPr>
        <w:tcW w:w="4788" w:type="dxa"/>
      </w:tcPr>
    </w:tcPrChange>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The tcPr element beneath the tcPrChange element specifies the set of table cell properties which were in place before the current revision to the document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcPrInner) is located in §A.1. *end note*]

17.4.71 tcW (Preferred Table Cell Width)

This element specifies the preferred width for this table cell. This preferred width is used as part of the table layout algorithm specified by the tblLayout element (§17.4.52; §17.4.53) - full description of the algorithm in the ST_TblLayout simple type (§17.18.87).

All widths in a table are considered preferred because:

- The table shall satisfy the shared columns as specified by the tblGrid element (§17.4.48)
- Two or more widths can have conflicting values for the width of the same grid column
- The table layout algorithm (§17.18.87) can require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type pct for this element shall be calculated relative to the overall width of the table.

If this element is omitted, then the cell width shall be of type auto.

[*Example*: Consider a WordprocessingML table defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  <w:tr>
    <w:tc>
      <w:tcPr>
        <w:tcW w:type="pct" w:w="33.3%"/>
      </w:tcPr>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:tcW w:type="pct" w:w="33.3%"/>
      </w:tcPr>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:tcW w:type="pct" w:w="33.3%"/>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
</w:tbl>
```

This table specifies that it has no preferred table width, but each cell must be exactly 33.3 percent of the overall table width. The resulting table would therefore be sized such that all columns are of the width of the maximum column, as follows:

	Hello world	

The text `Hello world` makes the middle cell larger, and the other two cells are increased in size to maintain the preferred widths of one-third of the overall table width. However, when the middle table cell requires a larger size to accommodate non-breaking text, that preference can be overridden as needed:

	Hello worlddddddddddddddddddddddddddddddddddddd	

In this case, the middle cell's long non breaking string caused the table to reach the text margins on the page, and therefore to override the preferred widths on the empty cells. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.4.72 textDirection (Table Cell Text Flow Direction)

This element specifies the direction of the text flow for this table cell.

If this element is omitted on a given table cell, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the table cell shall inherit the text flow settings from the parent section.

[*Example*: Consider a table with one cell in which all the table cell's text flow is oriented vertically, flowing from right to left horizontally within that cell:

This table cell would specify this text flow using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:textDirection w:val="rl" />
  </w:tcPr>
  …
</w:tc>
```

The textDirection element specifies via the `rl` value in the val attribute that the text flow is to be oriented vertically, with subsequent lines stacked from right to left. *end example*]

Attributes	Description

val (Direction of Text Flow)	Specifies the direction of the text flow for this object.
	
	[*Example*: Consider a document with a section in which text must be oriented vertically, flowing from left to right horizontally on the page. This setting requires the following WordprocessingML:
	
	`<w:sectPr>`
	`  …`
	`  <w:textDirection w:val="lr" />`
	`</w:sectPr>`
	
	The textDirection element specifies via the `lr` value in the val attribute that the text flow be oriented vertically, with subsequent lines stacked from left to right. *end example*]
	
	The possible values for this attribute are defined by the ST_TextDirection simple type (§17.18.93).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextDirection) is located in §A.1. *end note*]

17.4.73 tl2br (Table Cell Top Left to Bottom Right Diagonal Border)

This element specifies the border which shall be displayed on the top left side to bottom right diagonal within the current table cell.

If this element is omitted, then the top left to bottom right diagonal of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

[*Example*: Consider a table in which the first cell in the first row specifies a top left to bottom right diagonal cell border as follows:

R1C1	R1C2
R2C1	R2C2

This diagonal cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
       <w:tl2br w:val="double" w:sz="4" w:space="0" w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The tl2br element specifies a ½ point border of type double on the table cell's diagonal. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.74 top (Table Cell Top Border)

This element specifies the border which shall be displayed at the top of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net tblCellSpacing element value (§17.4.44;§17.4.43;§17.4.45) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the top of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a table in which the first cell in the first row specifies a top cell border , as follows:

R1C1	R1C2
R2C1	R2C2

This top cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
      <w:top w:val="thinThickThinSmallGap" w:sz="24" w:space="0"
w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The top element specifies a three point border of type `thinThinThickSmallGap`. *end example*]

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.75 top (Table Cell Top Margin Default)

This element specifies the amount of space which shall be left between the top extent of the cell contents and the top border of all table cells within the parent table. This setting can be overridden by the table cell top margin definition specified by the top element contained within the table cell's properties (§17.4.77).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a top margin is never specified in the style hierarchy, then this table shall have no top cell padding by default (excepting individual cell overrides).

[*Example*: Consider a two by two table in which the default table cell top margin is specified to be exactly 0.25 inches, as follows (marked with an arrow in the first table cell below):

R1C1	R2C1
R2C1	R2C2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This table property is specified using the following WordprocessingML markup:

```
<w:tbl>
  <w:tblPr>
    <w:tblCellMar>
      <w:top w:w="360" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  …
</w:tbl>
```

Every cell in the table has a default cell margin setting it to 360 twentieths of a point. *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.76 top (Table Top Border)

This element specifies the border which shall be displayed at the top of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the tcBorders element (§17.4.66) and the tblBorders element (§17.4.39;§17.4.38)

If this element is omitted, then the top of this table shall have the border specified by the associated table style. If no top border is specified in the style hierarchy, then this table shall not have a top border.

[*Example*: Consider a table in which the table properties specifies a top table border, as follows:

R1C1	R1C2
R2C1	R2C2

This top table border is specified using the following WordprocessingML:

```
<w:tbl>
  <w:tblPr>
    <w:tblBorders>
      <w:top w:val="thinThickThinMediumGap" w:sz="24" w:space="0"
w:color="D0D0D0" w:themeColor="accent3" w:themeTint="99"/>
    </w:tblBorders>
  </w:tblPr>
  …
</w:tbl>
```

The top element specifies a three point top table border of type thinThinThickMediumGap. *end example*].

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.77 top (Table Cell Top Margin Exception)

This element specifies the amount of space which shall be left between the top extent of the cell contents and the top border of a specific table cell within a table. This setting shall override the table cell top margin definition specified by the top element contained within the table properties (§17.4.75).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the top element contained within the table properties (§17.4.75).

[*Example*: Consider a table with two cells in which the first table cell's top margin is specified via an exception to be ten times larger (0.2 inches vs. 0.02 inches) than the other table cell margins:

This text fills the extents of the cell.
So does this

The first cell in the table would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:tcMar>
      <w:top w:w="288" w:type="dxa" />
    </w:tcMar>
  </w:tcPr>
</w:tc>
```

The first cell in this table has an exception applied to the table cell top cell margin setting it to 288 twentieths of a point (0.2 inches). *end example*]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.78 tr (Table Row)

This element specifies a single table row, which contains the table's cells. Table rows in WordprocessingML are analogous to HTML `tr` elements.

A tr element has one formatting child element, trPr (§17.4.81), which defines the properties for the row. Each unique property on the table row is specified by a child element of this element. As well, a table row can contain any row-level content, which allows for the use of table cells.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If a table cell does not include at least one child element other than the row properties, then this document shall be considered corrupt.

[*Example*: Consider a table consisting of a single table cell, which contains the text `Hello, world`:

Hello World

This table row's content is represented by the following WordprocessingML:

```
<w:tr>
  <w:tc>
    <w:tcPr>
      <w:tcW w:w="0" w:type="auto"/>
    </w:tcPr>
    <w:p>
      <w:r>
        <w:t>Hello, world</w:t>
      </w:r>
    </w:p>
  </w:tc>
</w:tr>
```

The tr element contains a single row-level element - in this case, a table cell. *end example*]

Attributes	Description
rsidDel (Revision Identifier for Table Row Deletion)	Specifies a unique identifier used to track the editing session when the row was deleted from the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Revision Identifier for Table Row)	Specifies a unique identifier used to track the editing session when the table row was added to the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidRPr (Revision Identifier for Table Row Glyph Formatting)	Specifies a unique identifier used to track the editing session when the glyph character representing the table row mark was last modified in the main document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidTr (Revision Identifier for Table Row Properties)	Specifies a unique identifier used to track the editing session when the table row's properties were last modified in this document. All rsid* attributes throughout this document with the same value, if present, shall indicate that those regions were modified during the same editing session (time between subsequent save actions). A producer can choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Row) is located in §A.1. *end note*]

17.4.79 tr2bl (Table Cell Top Right to Bottom Left Diagonal Border)

This element specifies the border which shall be displayed on the top right to bottom left diagonal within the current table cell.

If this element is omitted, then the top right to bottom left diagonal of this table cell shall not have a cell border, and its border can use the table's border settings as appropriate.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a table in which the second cell in the second row specifies a top right to bottom left diagonal cell border as follows:

R1C1	R1C2
R2C1	R2C2

This diagonal cell border is specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    …
    <w:tcBorders>
       <w:tr2bl w:val="double" w:sz="4" w:space="0" w:color="FF0000"/>
    </w:tcBorders>
  </w:tcPr>
  <w:p/>
</w:tc>
```

The tr2bl element specifies a ½ point border of type double on the table cell's diagonal. *end example*

This element's content model is defined by the common border properties definition in §17.3.4.

17.4.80 trHeight (Table Row Height)

This element specifies the height of the current table row within the current table. This height shall be used to determine the resulting height of the table row, which can be absolute or relative (depending on its attribute values).

If omitted, then the table row shall automatically resize its height to the height required by its contents (the equivalent of an hRule value of auto).

[*Example*: Consider the following WordprocessingML table:

Some text in R1C1.	

Examining the WordprocessingML for this table, the trHeight element is not specified, so the row heights are automatically determined by their contents (in the first row, the text Some text in R1C1.). If the first row shall be restricted to 0.1 inches high (144 twentieths of a point) regardless of its contents, that would be specified using the trHeight element as follows:

```
<w:trPr>
  <w:trHeight w:val="144" w:hRule="exact"/>
</w:trPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The resulting table row would be exactly 144 twentieths of a point high:

end example]

Attributes	Description

hRule (Table Row Height Type)	Specifies the meaning of the height specified for this table row.
	
	The meaning of the value of the val attribute is defined based on the value of the hRule attribute for this table row as follows:
	• If the value of hRule is `auto`, then the table row's height should be automatically determined based on the height of its contents. The h value is ignored.
	• If the value of hRule is `atLeast`, then the table row's height should be at least the value the h attribute.
	• If the value of hRule is `exact`, then the table row's height should be exactly the value of the h attribute.
	
	If this attribute is omitted, then its value shall be assumed to be `auto`.
	
	[*Example*: Consider the following paragraph containing a table row:
	
	`<w:tr>`
	`  <w:trPr>`
	`  <w:trHeight w:val="2189" w:hRule="atLeast"/>`
	`  </w:trPr>`
	`  …`
	`</w:tr>`
	
	The hRule attribute specifies a value of `atLeast`, so the table row is a minimum of 2189 twentieths of a point high regardless of its contents, since its val value is 2189 twentieths of a point. *end example*]
	
	The possible values for this attribute are defined by the ST_HeightRule simple type (§17.18.37).

val (Table Row Height)	Specifies the table row's height. This height is expressed in twentieths of a point. If this attribute is omitted, then its value shall be assumed to be `0`. The meaning of the value of the val attribute is defined based on the value of the hRule attribute for this table row as follows: • If the value of hRule is `auto`, then the table row's height should be automatically determined based on the height of its contents. This value is ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	• If the value of hRule is atLeast, then the table row's height should be at least the value of this attribute.
	• If the value of hRule is exact, then the table row's height should be exactly the value of this attribute.
	
	[*Example*: Consider the following table row:
	
	`<w:tr>`
	`  <w:trPr>`
	`  <w:trHeight w:val="2189" w:hRule="atLeast"/>`
	`  </w:trPr>`
	`  …`
	`</w:tr>`
	
	The val attribute specifies a value of 2189 twentieths of a point, so this table row is a minimum of 2189 twentieths of a point high regardless of its contents (growing if needed), since its hRule value is set to atLeast. *end example*]
	
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Height) is located in §A.1. *end note*]

17.4.81 trPr (Table Row Properties)

This element specifies the set of row-level properties applied to the current table row. Each unique property is specified by a child element of this element. These properties affect the appearance of all cells in the current row within the parent table, but can be overridden by individual cell-level properties, as defined by each property.

[*Example*: Consider the following WordprocessingML table:

The first row must have a table-row level property which specifies that it shall be restricted to 0.1 inches high (144 twentieths of a point) regardless of its contents, that would be specified using the trHeight element as follows:

```
<w:trPr>
  <w:trHeight w:val="144" w:hRule="exact"/>
  …
</w:trPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The trPr element specifies the set of table row properties applied to the current table row in the document, in this case a row height requirement using the trHeight element (§17.4.80). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrPr) is located in §A.1. *end note*]

17.4.82 trPr (Previous Table Row Properties)

This element specifies a previous set of table cell properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table cell property settings which were previously in place before a specific set of revisions by one author. Each unique property is specified by a child element of this element. These properties affect the appearance of all cells in the current row within the parent table, but can be overridden by individual cell-level properties, as defined by each property.

[*Example*: Consider the following WordprocessingML table:

This table has a row height for row one of exactly 0.1". If we change that to a row height of at least 0.1" with revision marking enabled, the table would appear as follows:

Some text in R1C1.	

The resulting WordprocessingML would be:

```
<w:tr>
  <w:trPr>
    <w:trHeight w:val="144"/>
    <w:trPrChange w:id="2" … >
      <w:trPr>
        <w:trHeight w:hRule="exact" w:val="144"/>
      </w:trPr>
    </w:trPrChange>
  </w:trPr>
  …
</w:tr>
```

The trPr element as a child of trPrChange specifies the set of table row properties which were in place before the current revision to the document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrPrBase) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.4.83 vAlign (Table Cell Vertical Alignment)

This element specifies the vertical alignment for text within the current table cell. The vertical alignment of this text is determined by the value of the val attribute.

[*Example*: Consider a table with a single cell with text vertically aligned to the bottom of the cell:

R1C1

This requirement would be specified using the following WordprocessingML:

```
<w:tc>
  <w:tcPr>
    <w:vAlign w:val="bottom" />
  </w:tcPr>
  <w:p>
    <w:r>
      <w:t>R1C1</w:t>
    </w:r>
  </w:p>
</w:tc>
```

The vAlign element specifies the vertical alignment of the cell contents, in the case, the `bottom` of the table cell. *end example*]

Attributes	Description
val (Vertical Alignment Setting)	Specifies the vertical alignment for text between the top and bottom margins of the parent container (page or table cell). [*Example*: Consider a region where the text must be vertically centered in the parent element. This would require a val value of `center`, in order to specify that all justification vertically must be centered relative to the parent. For a section, this setting would be specified as follows: `<w:vAlign w:val="center" />` The val attribute of `center` specifies that the content is centered relative to its container (in this case, the page). *end example*] The possible values for this attribute are defined by the ST_VerticalJc simple type (§17.18.101).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_VerticalJc) is located in §A.1. *end note*]

17.4.84 vMerge (Vertically Merged Cell)

This element specifies that this cell is part of a vertically merged set of cells in a table. The val attribute on this element determines how this cell is defined with respect to the previous cell in the table (i.e., whether this cell continues the vertical merge or starts a new merged group of cells).

If this element is omitted, then this cell shall not be part of any vertically merged grouping of cells, and any vertically merged group of preceding cells shall be closed. If a vertically merged group of cells do not span the same set of grid columns, then the document is non-conformant.

[*Example*: Consider a table with three rows and two columns with the last column completely vertically merged:

The second cell in the first row starts a vertical merge that is completed in the last cell, resulting in the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:vMerge w:val="restart"/>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tr>
  <w:tc>
    …
  </w:tc>
  <w:tc>
    …
  </w:tc>
  <w:tc>
    <w:tcPr>
      <w:vMerge w:val="continue"/>
    </w:tcPr>
    …
  </w:tc>
</w:tr>
<w:tr>
  <w:tc>
    …
  </w:tc>
  <w:tc>
    …
  </w:tc>
  <w:tc>
    <w:tcPr>
      <w:vMerge w:val="continue"/>
    </w:tcPr>
    …
  </w:tc>
</w:tr>
</w:tbl>
```

The vMerge element defines the cells that are vertically merged, and how each cell is merged together. *end example*]

Attributes	Description
val (Vertical Merge Type)	Specifies how the table cell is part of a vertically merged region. This determines whether the cell should join onto an existing grouping of merged cells if any exist, or start a new group of merged cells. Refer to the simple type definition for a full description of each type. If this attribute is omitted, its value shall be assumed to be `continue`. [*Example*: Consider a table cell where a vertical cell merge begins . This setting is represented as the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:tcPr> <w:vMerge w:val="restart"/> </w:tcPr> The attribute value of restart specifies that this element must start a new vertically merged region in this table. *end example*] The possible values for this attribute are defined by the ST_Merge simple type (§17.18.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VMerge) is located in §A.1. *end note*]

17.4.85 wAfter (Preferred Width After Table Row)

This element specifies the preferred width for the total number of grid columns after this table row as specified in the gridAfter element (§17.4.14). This preferred width is used as part of the table layout algorithm specified by the tblLayout element (§17.4.52; §17.4.53) - full description of the algorithm in the ST_TblLayout simple type (§17.18.87).

All widths in a table are considered preferred because:

- The table shall satisfy the shared columns as specified by the tblGrid element (§17.4.48)
- Two or more widths can have conflicting values for the width of the same grid column
- The table layout algorithm (§17.18.87) can require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type pct for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type auto.

[*Example*: Consider a WordprocessingML table row defined as follows:

```
<w:tr>
  <w:trPr>
    <w:gridAfter w:val="2"/>
    <w:wAfter w:type="dxa" w:w="1440"/>
  </w:trPr>
  …
</w:tr>
```

This table specifies that it has a preferred table width of 1440 twentieths of a point (one inch) for the two grid columns after the end of that row. The resulting table would therefore be sized such that that set of grid columns is one inch whenever possible, for example the second row in this table:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.86 wBefore (Preferred Width Before Table Row)

This element specifies the preferred width for the total number of grid columns before this table row as specified in the gridAfter element (§17.4.14). This preferred width is used as part of the table layout algorithm specified by the tblLayout element (§17.4.52; §17.4.53) - full description of the algorithm in the ST_TblLayout simple type (§17.18.87).

All widths in a table are considered preferred because:

- The table shall satisfy the shared columns as specified by the tblGrid element (§17.4.48)
- Two or more widths can have conflicting values for the width of the same grid column
- The table layout algorithm (§17.18.87) might require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type `pct` for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type `auto`.

[*Example*: Consider a WordprocessingML table row defined as follows:

```
<w:tr>
  <w:trPr>
    <w:gridBefore w:val="1"/>
    <w:wBefore w:type="dxa" w:w="1440"/>
  </w:trPr>
  …
</w:tr>
```

This table specifies that it has a preferred table width of `1440` twentieths of a point (one inch) for the grid column before the start of the row. The resulting table would therefore be sized such that that grid column is one inch whenever possible, for example the second row in this table:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

This element's content model is defined by the common table measurement definition in §17.4.87.

17.4.87 Table Measurement (CT_TblWidth)

This common complex type specifies a measurement to be used within a table. These properties contain two pieces of information:

- The type of measurement (percentage-based, absolute, or automatic)
- The value of the measurement

[*Example*: Consider the following table measurement:

```
<… w:type="pct" w:w="100%"/>
```

The type attribute specifies that the measurement is percentage-based, and the w attribute specifies the measurement is 100%. *end example*]

Attributes	Description
type (Table Width Type)	Specifies the units of the width property being defined by the parent element's w attribute. This property is used to define various properties of a table, including: cell spacing, preferred width, and table margins. If this attribute is omitted, then its value shall be assumed to be dxa (twentieths of a point). [*Example*: Consider a table with a table cell bottom cell spacing with a type of dxa, as follows: `<w:bottom … w:type="dxa" />` This type must therefore be used to interpret the width specified in the w attribute as a value in twentieths of a point. *end example*] If the value of the type attribute and the actual measurement specified by the w attribute are contradictory, the type specified by the type attribute shall be ignored. The possible values for this attribute are defined by the ST_TblWidth simple type (§17.18.90).
w (Table Width Value)	Specifies the value of the width property being defined by the parent element. This property is used to define various properties of a table, including: cell spacing, preferred widths, and table margins. If this attribute is omitted, then its value shall be assumed to be 0. [*Example*: Consider a table with a bottom margin with a width of 302, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:bottom w:w="302" w:type="dxa" /> The value in the w attribute must therefore be used to determine the width being specified in the context of the units specified in the type attribute. In this case, the type is twentieths of a point (dxa), so the width is 302 twentieths of a point (.2097 inches). *end example*] The possible values for this attribute are defined by the ST_MeasurementOrPercent simple type (§17.18.107).

[*Note*: The W3C XML Schema definition of this complex type's content model (CT_TblWidth) is located in §A.1. *end note*]

17.5 Custom Markup

Within a WordprocessingML document, semantic information may be supplied beyond the presentation information specified by ISO/IEC 29500. [*Example*: An invoice document might wish to specify that a particular sentence of text is a customer name, in order for that information to be easily extracted from the document without the need to parse the text using regular expression matching or similar. *end example*]

There are three distinct forms in which extra-standard semantics can be inserted into a WordprocessingML document, each with their own specific intended usage:

- Smart tags
- Custom XML markup
- Structured document tags (content controls)

The elements and attributes which define each of these forms is described in the following clauses.

17.5.1 Custom XML and Smart Tags

The first form of extra-standard semantics that can be embedded in a WordprocessingML document is represented by smart tags. Implementations can establish sets of smart tags that allow semantic labels to be added around an arbitrary run or set of runs within a document to provide information about the type of data contained within.

[*Example*: Consider the following text in a WordprocessingML document, with a smart tag around the stock symbol 'CNTS':

This is a stock symbol: CNTS

This text would translate to the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:p w:rsidR="00672474" w:rsidRDefault="00672474">
  <w:r>
    <w:t xml:space="preserve">This is a stock symbol: </w:t>
  </w:r>
  <w:smartTag w:uri="http://www.example.com"
     w:element="stockticker">
    <w:r>
      <w:t>CNTS</w:t>
    </w:r>
  </w:smartTag>
</w:p>
```

As shown above, the smart tag is delimited by the smartTag element, which surrounds the run (or runs) which contain the text which is part of the smart tag. *end example*]

The smartTag element has two required attributes:

- The uri attribute is used to specify the namespace, in terms of a vocabulary or classification scheme, of which the term specified for this smart tag is a member. [*Example*: In the above example, the smart tag specifies `http://www.example.com` to identify the classification scheme. *end example*]
- The element attribute, in combination with the uri attribute, specifies the classification for this smart tag. [*Example*: In the above example, the smart tag specifies the classification term `stockticker` from the identified namespace. *end example*]

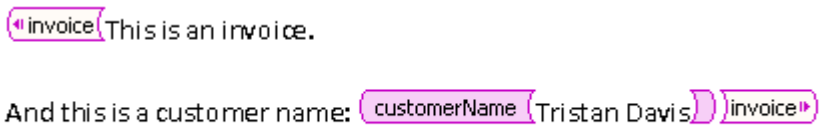
The second form of extra-standard semantics that can be embedded in a WordprocessingML document is custom XML markup. Custom XML markup allows the application of the XML elements defined in any schema syntax (XML Schema, NVDL, etc.) to be applied to the contents of a WordprocessingML document in two types of location: around a paragraph or set of paragraphs (at the block level); or around an arbitrary run or set of runs within a document (at the inline level) to provide semantics to that content within the context and structures defined by the associated schema definition.

The distinction between custom XML markup and smart tags is that custom XML markup is based on a schema, which may be specified using the attachedSchema element (§17.15.1.5). As a result, the custom XML elements can be validated against the schema. Also, as shown below, custom XML markup can be used at the block-level as well as on the inline (run) level.

[*Example*: Consider a simple XML Schema which defines two elements: a root element of invoice, and a child element of customerName - the first defining that this file's contents are an invoice, and the second specifying that the enclosed text as a customer's name:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



This output would translate to the following WordprocessingML markup:

```
<w:customXml w:uri="http://www.example.com/2006/invoice" w:element="invoice">
  <w:p>
    <w:r>
      <w:t>This is an invoice.</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t xml:space="preserve">And this is a customer name: </w:t>
    </w:r>
    <w:customXml w:uri="http://www.example.com/2006/invoice"
w:element="customerName">
      <w:r>
        <w:t>Tristan Davis</w:t>
      </w:r>
    </w:customXml>
  </w:p>
</w:customXml>
```

As shown above, each of the XML elements from the supplied XML schema is represented within the document output as a customXml element. *end example*]

A custom XML element in a document has two required attributes.

- The first is the uri attribute, whose contents specify the namespace of the custom XML element in the document. In the example above, the elements each belong to the `http://www.example.com/2006/invoice` namespace.
- The second is the element attribute, whose contents specify the name of the custom XML element at this location in the document. In the example above, the root element is called invoice and the child element is called customerName.

As well as the required information specified above, custom XML elements can also specify any number of attributes (as specified in the associated XML Schema) on the element. To add this information, the customXmlPr (properties on the custom XML element) specify one or more attr elements.

[*Example*: Using the example above, we can add a type attribute to the customerName element as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:customXml w:uri="http://www.example.com/2006/invoice"
w:element="customerName">
  <w:customXmlPr>
    <w:attr w:uri="http://www.example.com/2006/invoice" w:name="type"
w:val="individual"/>
  </w:customXmlPr>
  <w:r>
    <w:t>Tristan Davis</w:t>
  </w:r>
</w:customXml>
```

The resulting XML, as seen above, simply adds an attr element which specifies the attribute for the custom XML element. *end example*]…

17.5.1.1 attr (Custom XML Attribute)

This element specifies a custom XML attribute which shall be located on the parent custom XML element specified via the customXml element (§17.5.1.4;§17.5.1.5;§17.5.1.3; §17.5.1.6). The uri attribute can specify the Namespace of the custom XML attribute, and the name attribute shall specify the local name of the custom XML attribute. For any set of sibling attr elements, all the pairs of Namespace and local name shall be distinct.

[*Example*: Consider a custom XML element with the following properties:

```
<w:customXmlPr>
  <w:attr w:name="companyName" … />
  <w:attr w:name="companySymbol" … />
</w:customXmPr>
```

This set of custom XML properties specifies that the parent custom XML element must have two attributes associated with it, he first with a name of companyName, and the second with a name of companySymbol. *end example*]

Attributes	Description

name (Name)	Specifies the name of the current custom XML attribute or smart tag property.
	
	[*Example*: Consider a custom XML attribute which must have a name of companyName. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXmlPr>`
	`  <w:attr w:name="companyName" … />`
	`</w:customXmlPr>`
	
	The name attribute specifies that the name for this property must be companyName. *end example*]
	
	The possible values for this attribute are defined by the ST_XmlName simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.21).

uri (Namespace)	Specifies the namespace URI of the current custom XML attribute or smart tag property.
	
	If this attribute is omitted, the URI shall be assumed to be null (no associated URI).
	
	[*Example*: Consider a smart tag property which must have a namespace URI of `http://schemas.openxmlformats.org/2006/example`. This requirement would be specified using the following WordprocessingML:
	
	`<w:smartTagPr>`
	`  <w:attr w:uri="http://schemas.openxmlformats`
	`.org/2006/example" … />`
	`</w:smartTagPr>`
	
	The uri attribute specifies that the namespace for this property must be `http://schemas.openxmlformats.org/2006/example`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

val (Value)	Specifies the value of the current custom XML attribute or smart tag property.
	
	[*Example*: Consider a smart tag property which must have a value of `propertyValue`. This requirement would be specified using the following WordprocessingML:
	
	`<w:smartTagPr>`
	`  <w:attr … w:val="propertyValue" />`
	`</w:smartTagPr>`
	
	The val attribute specifies that the value for this property must be `propertyValue`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Attr) is located in §A.1. *end note*]

17.5.1.2 attr (Smart Tag Property)

This element specifies a single smart tag property which shall be located on the parent smart tag, specified via the smartTag element (§17.5.1.9). The attributes on this element shall be used to specify the contents of smart tag property.

[*Example*: Consider a smart tag with the following properties:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:smartTagPr>
  <w:attr w:name="attributeOne" … />
  <w:attr w:name="attributeTwo" … />
</w:smartTagPr>
```

This property bag specifies that the parent smart tag must have two properties associated with it, the first with a name of `attributeOne`, and the second with a name of `attributeTwo`. *end example*].

Attributes	Description

name (Name)	Specifies the name of the current custom XML attribute or smart tag property.
	
	[*Example*: Consider a custom XML attribute which must have a name of `companyName`. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXmlPr>`
	`  <w:attr w:name="companyName" … />`
	`</w:customXmlPr>`
	
	The name attribute specifies that the name for this property must be `companyName`. *end example*]
	
	The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).

uri (Namespace)	Specifies the namespace URI of the current custom XML attribute or smart tag property.
	
	If this attribute is omitted, the URI shall be assumed to be null (no associated URI).
	
	[*Example*: Consider a smart tag property which must have a namespace URI of `http://schemas.openxmlformats.org/2006/example`. This requirement would be specified using the following WordprocessingML:
	
	`<w:smartTagPr>`
	`  <w:attr w:uri="http://schemas.openxmlformats`
	`.org/2006/example" … />`
	`</w:smartTagPr>`
	
	The uri attribute specifies that the namespace for this property must be `http://schemas.openxmlformats.org/2006/example`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

val (Value)	Specifies the value of the current custom XML attribute or smart tag property. [*Example*: Consider a smart tag property which must have a value of `propertyValue`. This requirement would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:smartTagPr> <w:attr … w:val="propertyValue" /> </w:smartTagPr> The val attribute specifies that the value for this property must be propertyValue. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Attr) is located in §A.1. *end note*]

17.5.1.3 customXml (Inline-Level Custom XML Element)

This element specifies the presence of a custom XML element around one or more inline level structures (runs, images, fields, etc.) within a paragraph. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

[*Example*: Consider a custom XML element with the name firstName that must be located around a two text runs in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:p>
  <w:customXml w:element="firstName" … >
    <w:r>
      …
    </w:r>
    <w:r>
      …
    </w:r>
  </w:customXml>
  …
</w:p>
```

The customXml element specifies that the name of the custom XML element is firstName, and the custom XML element contains a two text runs (it is an inline-level element). *end example*]

Attributes	Description
element (Element name)	Specifies the name of the current custom XML element or smart tag within the document. [*Example*: Consider a custom XML element which must have a name of companyName. This requirement would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:customXml w:element="companyName" … > … </w:customXml> The element attribute specifies that the name for this element must be companyName. *end example*] The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).
uri (Custom XML Markup Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). [*Example*: Consider a custom XML element which must have a namespace URI of urn:customXmlExample. This requirement would be specified using the following WordprocessingML: <w:customXml … w:uri="urn:customXmlExample" > … </w:customXml> The uri attribute specifies that the namespace for this element must be urn:customXmlExample. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomXmlRun) is located in §A.1. *end note*]

17.5.1.4 customXml (Cell-Level Custom XML Element)

This element specifies the presence of a custom XML element around a single table cell. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

[*Example*: Consider a custom XML element with the name company that must be located around a single table cell in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tr>
  <w:customXml w:element="company" … >
    <w:tc>
      …
    </w:tc>
  </w:customXml>
  …
</w:tr>
```

The customXml element specifies that the name of the custom XML element is company, and the custom XML element contains a single table cell (it is a cell-level element). *end example*]

Attributes	Description

element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document.
	
	[*Example*: Consider a custom XML element which must have a name of companyName. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXml w:element="companyName" … >`
	`  …`
	`</w:customXml>`
	
	The element attribute specifies that the name for this element must be companyName. *end example*]
	
	The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).

uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag.
	
	If this attribute is omitted, the URI shall be assumed to be null (no associated URI).
	
	[*Example*: Consider a custom XML element which must have a namespace URI of urn:customXmlExample. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXml … w:uri="urn:customXmlExample" >`
	`  …`
	`</w:customXml>`
	
	The uri attribute specifies that the namespace for this element must be urn:customXmlExample. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomXmlCell) is located in §A.1. *end note*]

17.5.1.5 customXml (Row-Level Custom XML Element)

This element specifies the presence of a custom XML element around a single table row. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

[*Example*: Consider a custom XML element with the name `invoiceItem` that must be located around a single table row in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:tbl>
  <w:customXml w:element="invoiceItem" … >
    <w:tr>
      …
    </w:tr>
  </w:customXml>
  …
</w:tbl>
```

The customXml element specifies that the name of the custom XML element is `invoiceItem`, and the custom XML element contains a single table row (it is a row-level element). *end example*]

Attributes	Description

element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document.
	
	[*Example*: Consider a custom XML element which must have a name of `companyName`. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXml w:element="companyName" … >`
	`…`
	`</w:customXml>`
	
	The element attribute specifies that the name for this element must be `companyName`. *end example*]
	
	The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).

uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). [*Example*: Consider a custom XML element which must have a namespace URI of `urn:customXmlExample`. This requirement would be specified using the following

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	WordprocessingML:
	
	`<w:customXml … w:uri="urn:customXmlExample" >`
	`…`
	`</w:customXml>`
	
	The uri attribute specifies that the namespace for this element must be `urn:customXmlExample`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomXmlRow) is located in §A.1. *end note*]

17.5.1.6 customXml (Block-Level Custom XML Element)

This element specifies the presence of a custom XML element around one or more block level structures (paragraphs, tables, etc.). The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

[*Example*: Consider a custom XML element with the name `address` that must be located around a single paragraph in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:body>
  <w:customXml w:element="address" … >
    <w:p>
      …
    </w:p>
  </w:customXml>
  …
</w:body>
```

The customXml element specifies that the name of the custom XML element is `address`, and the custom XML element contains a single paragraph (it is a block-level element). *end example*]

Attributes	Description
element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document. [*Example*: Consider a custom XML element which must have a name of `companyName`. This requirement would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:customXml w:element="companyName" … > … </w:customXml> The element attribute specifies that the name for this element must be companyName. *end example*] The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).
uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). [*Example*: Consider a custom XML element which must have a namespace URI of urn:customXmlExample. This requirement would be specified using the following WordprocessingML: <w:customXml … w:uri="urn:customXmlExample" > … </w:customXml> The uri attribute specifies that the namespace for this element must be urn:customXmlExample. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomXmlBlock) is located in §A.1. *end note*]

17.5.1.7 customXmlPr (Custom XML Element Properties)

This element specifies the set of properties which shall be applied to the parent custom XML element.

[*Example*: Consider a custom XML element with the following properties specified:

```
<w:customXmlPr>
  <w:placeholder w:val="[Fill in your name]"/>
  <w:attr w:name="status" w:val="draft"/>
</w:customXmlPr>
```

This custom XML element specifies two properties: the presence of placeholder text via the placeholder element (§17.5.1.8) and a single custom XML attribute via the attr element (§17.5.1.1). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomXmlPr) is located in §A.1. *end note*]

17.5.1.8 placeholder (Custom XML Element Placeholder Text)

This element specifies the placeholder text which shall be displayed in place of this custom XML element when the contents of this custom XML markup are empty (i.e. there are no runs of text within the current custom XML element). If this custom XML element does contain run content, then this text shall not be displayed.

The val attribute stores the string of text which shall be displayed as the placeholder text. This string can be displayed in any font face/size desired by the hosting application.

[*Example*: Consider a custom XML element with the following properties specified:

```
<w:customXmlPr>
  <w:placeholder w:val="[Fill in your name]"/>
  <w:attr w:name="status" w:val="draft"/>
</w:customXmlPr>
```

The placeholder element specifies that this custom XML element must display the text contents `[Fill in your name]` whenever there is no run content within the parent custom XML element. For example, if the custom XML element was specified as follows:

```
<w:customXml>
  <w:customXmlPr>
    <w:placeholder w:val="[Fill in your name]"/>
  </w:customXmlPr>
  <w:p/>
</w:customXml>
```

This custom XML element has no run content and the placeholder text would be displayed. However, if there is run content, as follows:

```
<w:customXml>
  <w:customXmlPr>
    <w:placeholder w:val="[Fill in your name]"/>
  </w:customXmlPr>
  <w:p>
    <w:r>
      <w:t>Name</w:t>
    </w:r>
  </w:p>
</w:customXml>
```

This custom XML element now contains run content, and the placeholder text must not be displayed. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.1.9 smartTag (Inline-Level Smart Tag)

This element specifies the presence of a smart tag around one or more inline structures (runs, images, fields, etc.) within a paragraph. The attributes on this element shall be used to specify the name and namespace URI of the current smart tag.

[*Example*: Consider a smart tag with the name `firstName` that must be located around a two text runs in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:smartTag w:element="firstName" … >
    <w:r>
      …
    </w:r>
    <w:r>
      …
    </w:r>
  </w:smartTag>
  …
</w:p>
```

The smartTag element specifies that the name of the smart tag is `firstName`, and the smart tag contains a two text runs (it is an inline-level smart tag). *end example*]

Attributes	Description

element (Smart Tag Name)	Specifies the name of the current custom XML element or smart tag within the document.
	
	[*Example*: Consider a custom XML element which must have a name of `companyName`. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXml w:element="companyName" … >`
	`…`
	`</w:customXml>`
	
	The element attribute specifies that the name for this element must be `companyName`. *end example*]
	
	The possible values for this attribute are defined by the ST_XmlName simple type (§22.9.2.21).

uri (Smart Tag Namespace)	Specifies the namespace URI of the current custom XML element or smart tag.
	
	If this attribute is omitted, the URI shall be assumed to be null (no associated URI).
	
	[*Example*: Consider a custom XML element which must have a namespace URI of `urn:customXmlExample`. This requirement would be specified using the following WordprocessingML:
	
	`<w:customXml … w:uri="urn:customXmlExample" >`
	`…`
	`</w:customXml>`
	
	The uri attribute specifies that the namespace for this element must be `urn:customXmlExample`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagRun) is located in §A.1. *end note*]

17.5.1.10 smartTagPr (Smart Tag Properties)

This element specifies the set of properties which shall be applied to the parent smart tag.

[*Example*: Consider a smart tag with the following properties specified:

```
<w:smartTagPr>
  <w:attr w:name="date" w:val="01/01/2006"/>
  <w:attr w:name="status" w:val="draft"/>
</w:smartTagPr>
```

This smart tag specifies two properties: the presence of two smart tag properties via the attr element (§17.5.1.2). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagPr) is located in §A.1. *end note*]

17.5.2 Structured Document Tags

The final form of extra-standard semantics that can be embedded in a WordprocessingML document is represented by structured document tags (SDTs).

As shown above, smart tags and custom XML markup each provide a facility for embedding extra-standard semantics into the document: smart tags, via the ability to provide a basic namespace/name for a run or set of runs within a documents; and custom XML markup, via the ability to tag the document with XML elements and attributes specified by any XML Schema file.

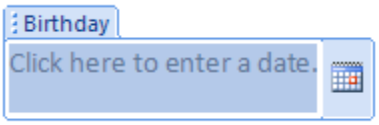
However, each of these techniques, while they each provide a way to add the desired semantic information, does not provide a way to affect the presentation or interaction within the document. To bridge these two worlds, structured document tags allow both the specification of extra-standard semantics as well as the ability to influence the presentation of that data in the document.

This means that the implementation can define the semantics and context of the tag, but can then use a rich set of pre-defined properties to define its behavior and appearance within the WordprocessingML document's presentation.

[*Example*: Consider a region which should be tagged with the semantic of "birthday", for the user to enter their date or birth into the document. Ideally, this region would also utilize a date picker to allow the user to enter the date from a calendar:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



This content would be specified using the following WordprocessingML:

```
<w:sdt>
  <w:sdtPr>
    <w:alias w:val="Birthday"/>
    <w:id w:val="8775518"/>
    <w:placeholder>
      <w:docPart w:val="DefaultPlaceholder_22479095"/>
    </w:placeholder>
    <w:showingPlcHdr/>
    <w:date>
      <w:dateFormat w:val="M/d/yyyy"/>
      <w:lid w:val="EN-US"/>
    </w:date>
  </w:sdtPr>
  <w:sdtContent>
    <w:p>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="PlaceholderText"/>
        </w:rPr>
        <w:t>Click here to enter a date…</w:t>
      </w:r>
    </w:p>
  </w:sdtContent>
</w:sdt>
```

end example]

As shown above, each of the structured document tags in the WordprocessingML file is represented using the sdt element.

Within a structured document tag, there are two child elements which contain the definition and the content of this SDT. The first of these is the sdtPr element, which contains the set of properties specified for this structured document tag. The second is the sdtContent element, which contains all the content which is contained within this structured document tag.

17.5.2.1 alias (Friendly Name)

This element specifies the friendly name associated with the current structured document tag. The string representing the friendly name shall be stored on this element's val attribute.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, then no friendly name shall be associated with the given structured document tag.

[*Example*: Consider the following properties on a structured document tag:

```
<w:sdtPr>
  <w:alias w:val="Birthday"/>
  …
</w:sdtPr>
```

This set of properties specifies via the alias element that the friendly name for the nearest ancestor structured document tag must be `Birthday`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.2.2 bibliography (Bibliography Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be of type `bibliography`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This setting does not require or imply that the contents of the structured document tag shall contain only a field of type BIBLIOGRAPHY, it shall only be used to specify that the structured document tag is of this kind, which can be used by an application as desired.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:bibliography/>
  </w:sdtPr>
  …
</w:sdt>
```

The bibliography element in this structured document tag's properties specify that the type of structured document tag is bibliography. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.3 calendar (Date Picker Calendar Type)

This element specifies the calendar which shall be displayed for the current date picker structured document tag, if a user interface is present for the structured document tag. The calendar information is stored on this element's val attribute.

If this element is omitted, then the calendar shall be gregorian.

[*Example*: Consider the following structured document tag properties:

```
<w:sdtPr>
  <w:date w:fullDate="2006-01-01T06:30:00Z">
    <w:calendar w:val="gregorian"/>
  </w:date>
</w:sdtPr>
```

The calendar element specifies that the calendar information for a calendar which might be displayed in the document must be the Gregorian calendar format (gregorian). *end example*]

Attributes	Description
val (Calendar Type Value)	Specifies a type of calendar, the use of which is determined by the parent XML element. If this attribute is omitted, then the calendar type shall be gregorian. [*Example*: Consider the following WordprocessingML for a document containing a structured document tag: <w:sdtPr>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:date … > <w:calendar w:val="japan"/> </w:date> </w:sdtPr> The val attribute value of japan specifies that the Japanese Emperor Era calendar must be used; in this case, it is used for the calendar displayed for a date structured document tag. *end example*] The possible values for this attribute are defined by the ST_CalendarType simple type (§22.9.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalendarType) is located in §A.1. *end note*]

17.5.2.4 citation (Citation Structured Document Tag)

This element specifies that the nearest ancestorstructured document tag shall be of type `citation`.

This setting does not require or imply that the contents of the structured document tag shall contain only a field of type `CITATION`, it shall only be used to specify that the structured document tag is of this type, which can be used by an application as desired.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:citation/>
  </w:sdtPr>
  …
</w:sdt>
```

The citation element in this structured document tag's properties specify that the type of structured document tag is `citation`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.5 comboBox (Combo Box Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a combo box when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The child elements of this element specify choices which shall be presented to the user

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag shall satisfy the following restraints or the document shall be considered non-conformant:

- The contents shall only be contain a single run (one set of formatting properties)
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:comboBox>
      …
    </w:comboBox>
  </w:sdtPr>
  …
</w:sdt>
```

The comboBox element in this structured document tag's properties specify that the type of structured document tag is a combo box. *end example*]

Attributes	Description
lastValue (Combo Box Last Saved Value)	Specifies the value associated with the current display text for the combo box structured document tag. If this structured document tag is not mapped to XML using the dataBinding element (§17.5.2.6), then this attribute shall be ignored. If this structured document tag is mapped to XML, it shall be used to determine whether the current display text in the combo box structured document tag shall be retained when the document is opened, as follows: • When the XML mapping is created, the content in the custom XML data is retrieved • If this content has an associated list item (matching its value attribute), then the corresponding display text shall be displayed in the structured document tag • If no list item exists, this content shall be matched against the lastValue attribute value. If the values match, the current display text shall be retained. If the values do not match, the current custom XML data content shall be the new display text (since no match exists in the combo box list items) [*Example*: Consider a combo box structured document tag defined as follows: `<w:sdt>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:sdtPr> <w:dataBinding … /> <w:comboBox w:lastValue="2"/> </w:sdtPr> <w:sdtContent> <w:r> <w:t>Hello, world</w:t> </w:r> </w:sdtContent> </w:sdt> The current run content of the structured document tag reads Hello, world. When this document is opened, if the current value of the associated custom XML data is 2, the matching lastValue attribute specifies that the contents of the combo box must continue to be the current display text of the combo boxeven though there is no listItem whose value is 2 (and normally, the content of the structured document tag would be set to 2. Essentially, this attribute specifies a listItem whose value is 2 and whose displayText is Hello, world (the current structured document tag contents). *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtComboBox) is located in §A.1. *end note*]

17.5.2.6 dataBinding (XML Mapping)

This element specifies the information that shall be used to establish a mapping between the nearest ancestor structured document tag and an XML element stored within a Custom XML Data part in the current WordprocessingML document.

If this element is omitted, then no XML mapping shall be associated with the current structured document tag. If the nearest ancestor structured document tag is of type rich text or document part gallery, then this property shall be ignored.

If this element is present and the nearest ancestor structured document tag is not of a rich text type, then the current value of the structured document tag shall be determined by finding the XML element (if any) which is determined by the attributes on this element. If this information does not result in an XML element, then the application can use any algorithm desired to find the closest available match. If this information does result in an XML element, then the contents of that element shall be used to replace the current run content within the document.

[*Example*: Consider the following structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    <w:dataBinding w:xpath="/root/name/first" … />
    <w:text/>
  </w:sdtPr>
  <w:sdtContent>
    <w:r>
      <w:t>old text</w:t>
    </w:r>
  </w:sdtContent>
</w:sdt>
```

This structured document tag specifies that it contains only plain text via the text element, and that it must be mapped to the element in the first Custom XML Data part which contains an element that matches the XPath expression `/root/name/first`. When that element is located, its contents must replace the existing run content in the document (for example, if its contents are `new text`, then the contents of the run for this structured document tag must be `new text` when the document is displayed. *end example*]

Attributes	Description

prefixMappings (XML Namespace Prefix Mappings)	Specifies the set of prefix mappings which shall be used to interpret the XPath expression specified on the xpath attribute when the XPath expression is evaluated against the custom XML data parts in the current document.
	
	This attribute's value shall be specified using the following syntax: `xmlns:prefix='namespace'`, where `prefix` is the namespace prefix to be mapped, and `namespace` is the namespace to be mapped to the current prefix. Each prefix mapping shall be delimited by one or more whitespace characters in the attribute's contents.
	
	If this attribute is omitted, then the prefix mappings specified on each of the custom XML data parts itself shall be used to evaluate the given XPath expression.
	
	[*Example*: Consider the following structured document tag properties:
	
	`<w:sdtPr>`
	`<w:dataBinding w:xpath="//ns0:book"`
	`w:prefixMappings="xmlns:ns0=`
	`'http://example.com/example'"/>`
	`<w:text/>`
	`</w:sdtPr>`
	
	This structured document tag specifies that it contains an XML mapping, and that mapping's prefixMapping attribute must signify that the set of namespace prefix mappings to be used to evaluate the xpath attribute value must be `xmlns:ns0='http://example.com/example'`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

storeItemID (Custom XML Data Storage ID)	Specifies the custom XML data identifier for the custom XML data part which shall be used to evaluate the given XPath expression. The *custom XML data identifier*, specified using the storeItemID attribute of the dataStoreItem element (§22.5.2.1) on the Custom XML Data Properties part is a string that uniquely identifies a particular custom XML data part in a WordprocessingML document (as multiple parts can have the same namespace for their root element).
	
	If specified, then the XPath expression specified on the xpath attribute shall only be evaluated against the custom XML data part whose properties part has a matching custom XML data identifier. If no custom XML data part exists with a matching identifier, then the XML mapping shall not be connected.
	
	If omitted, then the XPath expression shall be evaluated against each custom XML data part in turn until the given XPath expression is resolved to an XML element.
	
	[*Example*: Consider the following structured document tag properties:
	
	`<w:sdtPr>`
	`  <w:dataBinding w:xpath="//ns0:book"`
	`w:storeItemID="testXmlPart"`
	`/>`
	`  <w:text/>`
	`</w:sdtPr>`
	
	This structured document tag specifies that it contains an XML mapping, and that mapping must only be evaluated against the custom XML part whose identifier is equal to `testXmlPart` (if one exists). *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

xpath (XPath)	Specifies the XPath expression that shall be evaluated to find the custom XML node that is mapped to the nearest ancestor structured document tag. This XPath expression shall be specified using the syntax defined in the XML Path Language (XPath) Version 1.0 specification (see Annex A for bibliographic reference information).
	
	[*Example*: Consider the following structured document tag properties:
	
	`<w:sdtPr>`
	`  <w:dataBinding w:xpath="//ns0:book"`
	`w:prefixMappings="xmlns:ns0=`
	`'http://example.com/example'"/>`
	`  <w:text/>`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This structured document tag specifies that it contains an XML mapping, and that mapping's xpath attribute must signify that the XPath expression to be evaluated must be `//ns0:book`. Because the prefixMapping attribute is also specified, those prefix mappings must be used to evaluate this XPath expression. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataBinding) is located in §A.1. *end note*]

17.5.2.7 date (Date Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a date picker when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The child elements of this element specify how the dates in this structured document tag shall be stored in any mapped custom XML data and displayed in the document
- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag shall satisfy the following restraints or the document shall be considered non-conformant:

- The contents shall only be contain a single run (one set of formatting properties)
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table cell

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:date>
      …
    </w:date>
  </w:sdtPr>
  …
</w:sdt>
```

The date element in this structured document tag's properties specifies that the type of structured document tag is a date picker. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

fullDate (Last Known Date in XML Schema DateTime Format)	Specifies the full date and time last entered into the nearest ancestorstructured document tag using the standard XML Schema DateTime syntax.
	
	[*Note*: This cache is used because the date display mask stored on the dateFormat element (§17.5.2.8) might not contain all of the information about the date, which might be needed if the date display mask is later changed. *end note*]
	
	If this attribute is specified, then the current fullDate attribute shall be used to populate the run content of the nearest ancestor structured document tag by filtering it through the date display mask specified in the dateFormat element, if one is present.
	
	If this attribute is omitted, then the current display text shall be maintained when the document is displayed.
	
	[*Example*: Consider the following structured document tag properties:
	
	`<w:sdtPr>`
	`…`
	`<w:date w:fullDate="2006-01-01T05:30:00Z">`
	`…`
	`</w:date>`
	`</w:sdtPr>`
	
	The full XML Schema DateTime format for the current structured document tag is specified via the fullDate attribute value as `2006-01-01T05:30:00Z`. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtDate) is located in §A.1. *end note*]

17.5.2.8 dateFormat (Date Display Mask)

The element specifies the display format that shall be used to format any date entered into the nearest ancestor structured document tag in full DateTime format [*Example*: Through a user interface (a date picker), or through custom XML data associated with this structured document tag via the dataBinding element (§17.5.2.6). *end example*] before displaying it in the structured document tag's run content.

If this element is omitted, then the date shall be formatted using the standard date display mask for the language ID specified on the lid element (§17.5.2.20) if present, or the language ID of the run contents otherwise.

The date display mask specified in the val attribute shall be interpreted using the semantics specified in §17.16.4.1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following structured document tag properties:

```
<w:sdtPr>
  <w:date w:fullDate="2006-01-01T06:30:00Z">
    <w:dateFormat w:val="MM-YYYY"/>
  </w:date>
</w:sdtPr>
```

The full XML Schema DateTime format for the current structured document tag is specified via the fullDate attribute value as `2006-01-01T06:30:00Z`, and the date display mask is `MM-YYYY`, therefore the resulting date displayed in the document must be `01-2006` (the month and long year from the full date value, respectively). *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.5.2.9 docPart (Document Part Reference)

This element specifies the name of the document part that shall be displayed in the nearest ancestor structured document tag when its run contents are empty. If this element is specified, then a document part whose name element (§17.12.12) specifies a name matching the value of this element, and which belongs to the bbPlcHdr style shall be located to be used as the placeholder text for the nearest ancestorstructured document tag.

If no document part is located matching the criteria specified by this element, then five non-breaking spaces shall be used as the default placeholder text.

[*Example*: Consider a structured document tag defined as follows:

```
<w:sdt>
  <w:sdtPr>
    <w:placeholder>
      <w:docPart w:val="DefaultPlaceholder_22610170" />
    </w:placeholder>
    …
  </w:sdtPr>
  <w:sdtContent>
  …
  </w:sdtContent>
</w:sdt>
```

This structured document tag specifies through the docPart element that its placeholder text must be specified in the document part of style bbPlcHdr whose name is equal to DefaultPlaceholder_22610170. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's styleId.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.2.10 docPartCategory (Document Part Category Filter)

This element specifies the category of document parts that shall be used as the filter when determining the possible choices of document parts that are displayed for insertion into the nearest ancestor structured document tag. A document part *category* is a sub-classification within a given document part gallery which can be used to further categorize the parts in a given gallery. [*Example*: Gallery `custom1` might have categories of `Legal Clauses`, `Conformance Clauses`, etc. *end example*]. The category which shall be used as a filter is stored in this element's val attribute.

If this element is omitted, then the nearest ancestor structured document tag shall display all document parts in the specified gallery regardless their specified category. If this element is present, but no document parts of the specified gallery and category combination are located by the application, then no document parts shall be displayed (i.e. the application shall not fall back to showing document parts in all categories in the specified gallery).

[*Example*: Consider the following properties for a structured document tag:

```
<w:sdtPr>
  <w:docPartList>
    <w:docPartGallery w:val="custom1"/>
    <w:docPartCategory w:val="Legal Clauses"/>
  </w:docPartList>
</w:sdtPr>
```

This structured document tag specifies that it must present a selection of document parts for insertion via the docPartList element (§17.5.2.12) , and those document parts must only be the parts which are in the `custom1` gallery via the docPartType element (§17.5.2.11), and within that gallery, only the document parts which are in a category called `Legal Clauses` via this element. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.2.11 docPartGallery (Document Part Gallery Filter)

This element specifies the gallery of document parts that shall be used as the filter when determining the possible choices of document parts that are displayed for insertion into the nearest ancestor structured document tag. A document part *gallery* is a classification of document parts, which might then be subdivided into categories. [*Example*: A gallery with a name of `custom1` might have categories of `Legal Clauses`, `Conformance Clauses`, etc. *end example*]. The gallery which shall be used is stored in this element's val attribute.

If this element is omitted, then the nearest ancestor structured document tag shall display all document parts in its default gallery. If this element is present, but no document parts of the specified gallery are located by the application, then document parts in the default gallery shall be displayed (i.e. the application shall behave as if the value was omitted).

[*Example*: Consider the following properties for a structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdtPr>
  <w:docPartList>
    <w:docPartGallery w:val="custom1"/>
  </w:docPartList>
</w:sdtPr>
```

This structured document tag specifies that it must present a selection of document parts for insertion via the docPartList element (§17.5.2.12), and those document parts must only be the parts which are in the custom1 gallery via this element. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's styleId.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.2.12 docPartList (Document Part Gallery Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be of a document part gallery type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This setting does not require or imply that the contents of the structured document tag shall contain only the exact contents of a document part of the specified gallery and category which is present on the current machine, it shall only be used to specify that the structured document tag is of this kind, which shall be used by an application to present the possible list of choices for insertion into the nearest ancestor structured document tag.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:docPartList>
      …
    </w:docPartList>
  </w:sdtPr>
  …
</w:sdt>
```

The docPartList element in this structured document tag's properties specifies that the type of structured document tag is a document part gallery. The child elements must specify the gallery and category filters for this list, if any. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtDocPart) is located in §A.1. *end note*]

17.5.2.13 docPartObj (Built-In Document Part Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be of a document part type.

This setting does not require or imply that the contents of the structured document tag shall contain only the exact contents of a document part of the specified gallery and category which is present on the current machine, it shall only be used to specify that the structured document tag is of this kind, which shall be used by an application to present the possible list of choices for insertion into the nearest ancestor structured document tag.

This element differs from the docPartList element (§17.5.2.12) in that it can be used to semantically tag a set of block-level objects in a WordprocessingML document without requiring the ability to specify a category and gallery of objects which can be swapped with it via the user interface.

[*Example*: Consider the following structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    …
    <w:docPartObj>
      …
    </w:docPartObj>
  </w:sdtPr>
  …
</w:sdt>
```

The docPartObj element in this structured document tag's properties specify that the type of structured document tag is a document part. The child elements must specify the gallery and category semantics for this part, if any. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtDocPart) is located in §A.1. *end note*]

17.5.2.14 docPartUnique (Built-In Document Part)

This element specifies that this structured document tag is being used to encapsulate a built-in document part (i.e. this element appears as a child element of the docPartObj element).

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:docPartObj>
      …
      <w:docPartUnique/>
    </w:docPartObj>
  </w:sdtPr>
  …
</w:sdt>
```

The docPartUnique element in this structured document tag's properties specify that the type of structured document tag is a container for a document part. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.5.2.15 dropDownList (Drop-Down List Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a drop-down list when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- The contents shall not be editable when displayed by a hosting application regardless of the locking settings
- The child elements of this element specify choices which shall be displayed in a standard drop-down list format

As well, the structured document tag shall satisfy the following restraints or the document shall be considered non-conformant:

- The contents shall only be contain a single run (one set of formatting properties)
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:dropDownList>
      …
    </w:dropDownList>
  </w:sdtPr>
  …
</w:sdt>
```

The dropDownList element in this structured document tag's properties specify that the type of structured document tag is a drop-down list. *end example*]

Attributes	Description
lastValue (Drop-down List Last Saved Value)	Specifies the value associated with the current display text for the drop-down list structured document tag. If this structured document tag is not mapped to XML using the dataBinding element (§17.5.2.6), then this attribute shall be ignored. If this structured document tag is mapped to XML, it shall be used to determine whether the current display text in the combo box structured document tag shall be retained when the document is opened, as follows: • When the XML mapping is created, the content in the custom XML data is retrieved • If this content has an associated list item (matching its value attribute), then the corresponding display text shall be displayed in the structured document tag • If no list item exists, this content shall be matched against the lastValue attribute value. If the values match, the current display text shall be retained. If the values do not match, the current custom XML data content shall be the new display text (since no match exists in the combo box list items) [*Example*: Consider a drop-down list structured document tag defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:sdt> <w:sdtPr> <w:dataBinding … /> <w:dropDownList w:lastValue="2"/> </w:sdtPr> <w:sdtContent> <w:r> <w:t>Hello, world</w:t> </w:r> </w:sdtContent> </w:sdt> The current run content of the structured document tag reads Hello, world. When this document is opened, if the current value of the associated custom XML data is 2, the matching lastValue attribute specifies that the contents of the combo box must continue to be the current display text of the combo boxeven though there is no listItem whose value is 2 (and normally, the content of the structured document tag would be set to 2. Essentially, this attribute specifies a listItem whose value is 2 and whose displayText is Hello, world (the current structured document tag contents). *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtDropDownList) is located in §A.1. *end note*]

17.5.2.16 equation (Equation Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be of type equation.

This setting does not require or imply that the contents of the structured document tag shall contain only an equation or associated placeholder text, it shall only be used to specify that the structured document tag is of this kind, which can be used by an application as desired.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:equation/>
  </w:sdtPr>
  …
</w:sdt>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The equation element in this structured document tag's properties specify that the type of structured document tag is `equation`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.17 group (Group Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a restricted grouping when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The contents of this structured document tag shall not be editable when displayed by a hosting application regardless of the locking settings. This restriction can be superseded by any structured document tag contained within the group, as each structured document tag specifies the locking properties for its own content.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:group/>
  </w:sdtPr>
  …
</w:sdt>
```

The group element in this structured document tag's properties specify that the type of structured document tag is a restricted group. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.18 id (Unique ID)

This element specifies a unique numerical ID for the nearest ancestor structured document tag. This ID shall be persisted through multiple sessions (i.e. shall not be changed once specified).

If multiple structured document tags specify the same decimal number value for the id attribute, then the first structured document tag in the document shall maintain this original ID, and all subsequent structured document tags shall have new identifiers assigned to them when the document is opened.

If this element is omitted, then the nearest ancestor structured document tag shall have a new unique identifier assigned to it when the document is opened.

[*Example*: Consider the following structured document tag properties:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdtPr>
  <w:id w:val="8775518"/>
  …
</w:sdtPr>
```

This set of properties specifies via the val attribute on the id element that the ID for the parent structured document must be 8775518 (subject, of course, to the conflict management and resolution discussed above). *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.5.2.19 label (Structured Document Tag Label)

This element specifies the label identifier associated with the current structured document tag. The identifier representing the label shall be stored on this element's val attribute and is used to reference the unique identifier value of a structured document tag. The contents of the structured document tag resolved by a specific unique identifier shall be used as the label content for the structured document tag that references that specific unique identifier of the structured document tag. If multiple instances of the label element are present, the labels referenced are ordered from most general to most specific. [*Example*: A form element for specifying country name might reference the label for these three items (in order): "Sender", "Home Address", and "Country". *end example*]

If this element is omitted or the value of the label identifier cannot be resolved, then no label shall be associated with the given structured document tag.

[*Example*: Consider the following two structured document tags where one structured document tag references another structured document tag as a label:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:sdt>
  <w:sdtPr>
    <w:id w:val="5" />
  </w:sdtPr>
  <w:sdtContent>
    <w:p>
      <w:r>
        <w:t>Name</w:t>
      </w:r>
    </w:p>
  </w:sdtContent>
</w:sdt>
…
<w:sdt>
  <w:sdtPr>
    <w:id w:val="6" />
    <w:label w:val="5" />
  </w:sdtPr>
  …
</w:sdt>
```

The label element specifies that the structured document tag with an identifier value of 6 uses the contents of the structured document tag with an identifier value of 5 as a label source. In this example, the label contents are “Name”. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element’s content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.5.2.20 lid (Date Picker Language ID)

This element specifies the language ID that shall be used for displaying a calendar for the current date picker structured document tag, if a user interface is present for the structured document tag.

If this element is omitted, then the language ID shall be the language ID of the run contents of the nearest ancestor structured document tag.

[*Example*: Consider the following structured document tag properties:

```
<w:sdtPr>
  <w:date w:fullDate="2006-01-01T06:30:00Z">
    <w:lid w:val="ja-JP"/>
  </w:date>
</w:sdtPr>
```

The calendar language ID for a calendar which can be displayed in the document must be the default calendar format for the Japanese (Japan) language format (`ja-JP`). *end example*]

Attributes	Description
val (Language Code)	Specifies an identifier for a specific language. This code is interpreted in the context of the parent XML element. [*Example*: Consider an object which must specify the English(Canada) language. That object would use an identifier of `en-CA` to specify this language. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lang) is located in §A.1. *end note*]

17.5.2.21 listItem (Combo Box List Item)

This element specifies a single list item within the parent combo box structured document tag. Each list item shall be displayed in the list displayed for the nearest ancestor structured document tag (if a user interface is present).

[*Example*: Consider the following combo box structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    <w:comboBox>
      <w:listItem w:displayText="Zero" w:value="0"/>
      <w:listItem w:displayText="One" w:value="1"/>
    </w:comboBox>
  </w:sdtPr>
  …
</w:sdt>
```

Each listItem element within the comboBox element specifies a single list item entry, in this case resulting in two list items within the parent combo box structured document tag. *end example*]

Attributes	Description

displayText (List Entry Display Text)	Specifies the text to display in the run content (as well as any supplied user interface) in place of the value attribute contents for this drop-down list entry.
	
	This value shall be used as follows:
	• If the nearest ancestor structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped the content of the value attribute, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed.
	• If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's run content in the document (this is the value that shall be shown in the document's WordprocessingML content).
	
	If this attribute is omitted, then the content of the value attribute shall be used as the display text for the current list item entry.
	
	[*Example*: Consider the following drop-down list structured document tag:
	
	`<w:sdt>`
	`  <w:sdtPr>`
	`  <w:dropDownList>`
	`  <w:listItem w:displayText="The Letter A" w:value="a"/>`
	`  <w:listItem w:displayText="The Letter B" w:value="b"/>`
	`  </w:dropDownList>`
	`  </w:sdtPr>`
	`  …`
	`</w:sdt>`
	
	The displayText attribute for the first entry is `The Letter A` and the second is `The Letter B`, therefore, these values are used to determine the display text if the nearest ancestor structured document tag is mapped to custom XML data in a custom XML data part. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

value (List Entry Value)	Specifies the value for the current list item entry.
	
	This value shall be used as follows:
	• If the nearest ancestor structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped to this value, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed.
	• If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's listItem attribute value.
	
	[*Example*: Consider the following combo box structured document tag:
	
	`<w:sdt>`
	`  <w:sdtPr>`
	`  <w:comboBox>`
	`  <w:listItem w:displayText="Zero" w:value="0"/>`
	`  <w:listItem w:displayText="One" w:value="1"/>`
	`  </w:comboBox>`
	`  </w:sdtPr>`
	`  …`
	`</w:sdt>`
	
	The value attribute for the first entry is `0` and the second is `1`, therefore, these values are used to determine the display text if the nearest ancestor structured document tag is mapped to custom XML data in a custom XML data part. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtListItem) is located in §A.1. *end note*]

17.5.2.22 listItem (Drop-Down List Item)

This element specifies a single list item within the parent drop-down list structured document tag. Each list item shall be displayed in the list displayed for the nearest ancestor structured document tag (if a user interface is present).

[*Example*: Consider the following combo box structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    <w:dropDownList>
      <w:listItem w:displayText="The Letter A" w:value="a"/>
      <w:listItem w:displayText="The Letter B" w:value="b"/>
    </w:dropDownList>
  </w:sdtPr>
  …
</w:sdt>
```

Each listItem element within the dropDownList element specifies a single list item entry, in this case resulting in two list items within the parent drop-down list structured document tag. *end example*]

Attributes	Description

displayText (List Entry Display Text)	Specifies the text to display in the run content (as well as any supplied user interface) in place of the value attribute contents for this drop-down list entry.
	
	This value shall be used as follows:
	• If the nearest ancestor structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped the content of the value attribute, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed.
	• If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's run content in the document (this is the value that shall be shown in the document's WordprocessingML content).
	
	If this attribute is omitted, then the content of the value attribute shall be used as the display text for the current list item entry.
	
	[*Example*: Consider the following drop-down list structured document tag:
	
	`<w:sdt>`
	`  <w:sdtPr>`
	`  <w:dropDownList>`
	`  <w:listItem w:displayText="The Letter A" w:value="a"/>`
	`  <w:listItem w:displayText="The Letter B" w:value="b"/>`
	`  </w:dropDownList>`
	`  </w:sdtPr>`
	`  …`
	`</w:sdt>`
	
	The displayText attribute for the first entry is `The Letter A` and the second is `The Letter B`, therefore, these values are used to determine the display text if the nearest ancestor structured document tag is mapped to custom XML data in a custom XML data part. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

value (List Entry Value)	Specifies the value for the current list item entry.
	
	This value shall be used as follows:
	• If the nearest ancestor structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped to this value, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed.
	• If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's listItem attribute value.
	
	[*Example*: Consider the following combo box structured document tag:
	
	`<w:sdt>`
	`  <w:sdtPr>`
	`  <w:comboBox>`
	`  <w:listItem w:displayText="Zero" w:value="0"/>`
	`  <w:listItem w:displayText="One" w:value="1"/>`
	`  </w:comboBox>`
	`  </w:sdtPr>`
	`  …`
	`</w:sdt>`
	
	The value attribute for the first entry is `0` and the second is `1`, therefore, these values are used to determine the display text if the nearest ancestor structured document tag is mapped to custom XML data in a custom XML data part. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtListItem) is located in §A.1. *end note*]

17.5.2.23 lock (Locking Setting)

This element specifies the set of behaviors that shall be applied to the contents of the nearest ancestor structured document tag when the contents of this document are edited by an application (whether through a user interface or directly). The type of locking applied to the structured document tag is specified via the value of the associated val attribute.

If this element is omitted, then the locking settings implied for the structured document tag shall be as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- If the structured document tag specifies that it is a group via the group element (§17.5.2.17), then the contents of the structured document tag shall be editable, but the entire tag can be deleted.
- For all other kinds, no locking settings shall be applied to the structured document tag.

[*Example*: Consider the following plain text structured document tag:

```
<w:sdt>
  <w:sdtPr>
    <w:lock w:val="sdtLocked"/>
    …
    <w:text/>
  </w:sdtPr>
  …
</w:sdt>
```

This plain text structured document tag's properties contain a lock element, specifying locking behaviors for the structured document tag. Since the locking val attribute value is `sdtLocked`, this locking setting must specify that the contents of the structured document tag can be edited, but the structured document tag itself must not be deleted from the document. *end example*]

Attributes	Description

val (Locking Type)	Specifies the type of locking which shall be applied to the nearest ancestor structured document tag.
	
	If this attribute is omitted, this its value shall be assumed to be `unlocked` (using the defaults stated above).
	
	[*Example*: Consider the following plain text structured document tag properties:
	
	`<w:sdtPr>`
	`  <w:lock w:val="contentLocked"/>`
	`  …`
	`  <w:text/>`
	`</w:sdtPr>`
	
	The val attribute value is `contentLocked`, therefore this locking setting shall specify that the contents of the structured document tag must not be edited, but the structured document tag itself can be deleted from the document. *end example*]
	
	The possible values for this attribute are defined by the ST_Lock simple type (§17.18.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lock) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.5.2.24 picture (Picture Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a picture when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The contents shall always be restricted to a single picture using the DrawingML (§20.1) syntax

As well, the structured document tag shall satisfy the following restraints or the document shall be considered non-conformant:

- The contents shall only be a single picture using the DrawingML (§20.1) syntax
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:picture/>
  </w:sdtPr>
  …
</w:sdt>
```

The text element in this structured document tag's properties specify that the type of structured document tag is a picture. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.25 placeholder (Structured Document Tag Placeholder Text)

This element specifies the placeholder text which should be displayed when this structured document tag's run contents are empty, the associated mapped XML element is empty as specified via the dataBinding element (§17.5.2.6) or the showingPlcHdr element (§17.5.2.39) is set in the structured document tag's properties. The placeholder text which shall be shown is itself specified via the child element docPart.

If this element is omitted, then five non-breaking spaces shall be used as the default placeholder text for this structured document tag.

[*Example*: Consider a structured document tag defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:sdt>
  <w:sdtPr>
    <w:placeholder>
      <w:docPart w:val="DefaultPlaceholder_22610170" />
    </w:placeholder>
    …
  </w:sdtPr>
  <w:sdtContent>
    …
  </w:sdtContent>
</w:sdt>
```

This structured document tag specifies through the placeholder element that its placeholder text must be specified in the document part of type `bbPlcHdr` whose name is equal to `DefaultPlaceholder_22610170`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Placeholder) is located in §A.1. *end note*]

17.5.2.26 richText (Rich Text Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a rich text box when displayed in the document.

If no type element (the xsd:choice block in the XML Schema fragment for the parent sdtPr element) is specified, then the nearest ancestor structured document tag shall be of type richText.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:richText/>
  </w:sdtPr>
  …
</w:sdt>
```

The richText element in this structured document tag's properties specify that the type of structured document tag is a rich text box. *end example*].

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.5.2.27 rPr (Run Properties For Structured Document Tag Contents)

This element specifies the set of run properties that shall be applied to the text entered into the nearest ancestor structured document tag in replacement of placeholder text. When placeholder text is present in a structured document tag, its formatting is often different than the desired underlying formatting, and this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

element specifies the formatting which shall be used for non-placeholder text contents when they are initially added to the control.

If this element is not present, the inserted is unformatted, as with any other run of text - it shall not inherit the properties of the placeholder text.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    <w:placeholder>
      <w:docPart w:val="TestPlaceholderDocPart"/>
    </w:placeholder>
    <w:showingPlcHdr/>
    <w:rPr>
      <w:rStyle w:val="UserName"/>
    </w:rPr>
    …
  </w:sdtPr>
  <w:sdtContent>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="PlaceholderText"/>
      </w:rPr>
      <w:t>[Type Your Name Here]</w:t>
    </w:r>
  </w:sdtContent>
</w:sdt>
```

This structured document tag specifies that its current contents are placeholder text via the showingPlcHdr element (§17.5.2.39), and that text has the `PlaceholderText` character style applied to it.

Now, assume that that style created grey shaded text (typical for placeholder text). This formatting would clearly not be desirable for any text entered into the structured document tag. Therefore, when this text is added, the rPr element in the sdtPr is used to store the formatting on the resulting text.

In this example, the text which initially populates the control shall be formatted with the `UserName` character style. *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.5.2.28 rPr (Structured Document Tag End Character Run Properties)

This element specifies the set of run properties which shall be applied to the character present to delimit the end of the structured document tag's contents. When these properties are applied, they shall be applied in addition to the run properties specified for the entire structured document tag via the rPr element (§17.5.2.27) stored in the tag's main property container.

If this element is not present, the inserted closing tag shall be formatting identically to the start tag.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    <w:placeholder>
      <w:docPart w:val="TestPlaceholderDocPart"/>
    </w:placeholder>
    <w:showingPlcHdr/>
    <w:rPr>
      <w:rStyle w:val="UserName"/>
    </w:rPr>
    …
  </w:sdtPr>
  <w:sdtEndPr>
    <w:rPr>
      <w:b/>
      <w:i/>
    </w:rPr>
  </w:sdtEndPr>
  <w:sdtContent>
    …
  </w:sdtContent>
</w:sdt>
```

The rPr elements under the tag's properties specify that this structured document tag specifies that its start character must have formatting in the character style `UserName`, and that the end character must have the formatting in the character style `UserName` as well as bold and italic direct formatting. *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

17.5.2.29 sdt (Block-Level Structured Document Tag)

This element specifies the presence of a structured document tag around one or more block-level structures (paragraphs, tables, etc.). The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the sdtPr and sdtContent elements, respectively.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a structured document tag with the friendly name `address` that must be located around a single paragraph in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:body>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="address"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:p>
        …
      </w:p>
    </w:sdtContent>
  </w:sdt>
  …
</w:body>
```

The sdt element specifies the structured document tag, the child sdtPr element contains the friendly name property set to `address`, and the sdtContent element contains a single paragraph (it is a block-level structured document tag). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtBlock) is located in §A.1. *end note*]

17.5.2.30 sdt (Row-Level Structured Document Tag)

This element specifies the presence of a structured document tag around a single table row. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the sdtPr and sdtContent elements, respectively.

[*Example*: Consider a structured document tag with the friendly name `invoiceItem` that must be located around a single table row in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tbl>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="invoiceItem"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:tr>
        …
      </w:tr>
    </w:sdtContent>
  </w:sdt>
  …
</w:tbl>
```

The sdt element specifies the structured document tag, the child sdtPr element contains the friendly name property set to `invoiceItem`, and the sdtContent element contains a single table row (it is a row-level structured document tag). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtRow) is located in §A.1. *end note*]

17.5.2.31 sdt (Inline-Level Structured Document Tag)

This element specifies the presence of a structured document tag around one or more inline-level structures (runs, DrawingML objects, fields, etc.) in the current paragraph. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the sdtPr and sdtContent elements, respectively.

[*Example*: Consider a structured document tag with the friendly name `firstName` that must be located around two runs in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="firstName"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:r>
        …
      </w:r>
      <w:r>
        …
      </w:r>
    </w:sdtContent>
  </w:sdt>
  …
</w:p>
```

The sdt element specifies the structured document tag, the child sdtPr element contains the friendly name property set to `firstName`, and the sdtContent element contains two runs (it is an inline-level structured document tag). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtRun) is located in §A.1. *end note*]

17.5.2.32 sdt (Cell-Level Structured Document Tag)

This element specifies the presence of a structured document tag around a single table cell. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the sdtPr and sdtContent elements, respectively.

[*Example*: Consider a structured document tag with the friendly name `company` that must be located around a single table cell in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tr>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="company"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:tc>
        …
      </w:tc>
    </w:sdtContent>
  </w:sdt>
  …
</w:tr>
```

The sdt element specifies the structured document tag, the child sdtPr element contains the friendly name property set to company, and the sdtContent element contains a single table cell (it is a cell-level structured document tag). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtCell) is located in §A.1. *end note*]

17.5.2.33 sdtContent (Cell-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around a single table cell. This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the dataBinding element (§17.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the showingPlcHdr element (§17.5.2.39), then this content can be updated with the placeholder text stored in the Glossary Document part

[*Example*: Consider a structured document tag with the friendly name company that must be located around a single table cell in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tr>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="company"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:tc>
        …
      </w:tc>
    </w:sdtContent>
  </w:sdt>
  …
</w:tr>
```

The sdtContent element contains a single table cell (it is an cell-level structured document tag content container). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtContentCell) is located in §A.1. *end note*]

17.5.2.34 sdtContent (Block-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around one or more block-level structures (paragraphs, tables, etc.). This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the dataBinding element (§17.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the showingPlcHdr element (§17.5.2.39), then this content can be updated with the placeholder text stored in the Glossary Document part

[*Example*: Consider a structured document tag with the friendly name address that must be located around a single paragraph in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:body>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="address"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:p>
        …
      </w:p>
    </w:sdtContent>
  </w:sdt>
  …
</w:body>
```

The sdtContent element contains a single paragraph (it is a block-level structured document tag content container). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtContentBlock) is located in §A.1. *end note*]

17.5.2.35 sdtContent (Row-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around a single table row.

[*Note*: Unlike other types of structured document tags, this type of structure document tag cannot show placeholder text or have mapped XML data, therefore it is never a cache. *end note*]

[*Example*: Consider a structured document tag with the friendly name `invoiceItem` that must be located around a single table row in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:tbl>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="invoiceItem"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:tr>
        …
      </w:tr>
    </w:sdtContent>
  </w:sdt>
  …
</w:tbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The sdtContent element contains a single table row (it is an row-level structured document tag content container). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtContentRow) is located in §A.1. *end note*]

17.5.2.36 sdtContent (Inline-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around one or more inline-level structures (runs, DrawingML objects, fields, etc.). This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the dataBinding element (§17.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the showingPlcHdr element (§17.5.2.39), then this content can be updated with the placeholder text stored in the Glossary Document part

[*Example*: Consider a structured document tag with the friendly name firstName that must be located around two runs in a WordprocessingML document. This requirement would be specified as follows in the WordprocessingML:

```
<w:p>
  <w:sdt>
    <w:sdtPr>
      <w:alias w:val="firstName"/>
    </w:sdtPr>
    <w:sdtContent>
      <w:r>
        …
      </w:r>
      <w:r>
        …
      </w:r>
    </w:sdtContent>
  </w:sdt>
  …
</w:p>
```

The sdtContent element contains two adjacent runs (it is an inline-level structured document tag content container). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtContentRun) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.5.2.37 sdtEndPr (Structured Document Tag End Character Properties)

This element specifies the properties which shall be applied to the physical character which delimits the end of a structured document tag.

[*Example*: Consider a structured document tag with the following properties specified for the end tag:

```
<w:sdtEndPr>
  <w:rPr>
    …
  </w:rPr>
</w:sdtEndPr>
```

This structured document tag specifies properties for its end character within the sdtEndPr element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtEndPr) is located in §A.1. *end note*]

17.5.2.38 sdtPr (Structured Document Tag Properties)

This element specifies the set of properties that shall be applied to the nearest ancestor structured document tag.

[*Example*: Consider a structured document tag with the following properties specified:

```
<w:sdtPr>
  <w:alias w:val="Birthday"/>
  <w:id w:val="8775518"/>
  <w:date>
    <w:dateFormat w:val="M/d/yyyy"/>
    <w:lid w:val="EN-US"/>
  </w:date>
</w:sdtPr>
```

This structured document tag specifies three properties: the a friendly name of `Birthday` via the alias element (§17.5.2.1), a unique ID of `8775518` via the id element (§17.5.2.18), and a structured document tag type of date picker via the date element (§17.5.2.7) which itself has a set of date-specific properties. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtPr) is located in §A.1. *end note*]

17.5.2.39 showingPlcHdr (Current Contents Are Placeholder Text)

This element specifies whether the content of the sdtContent element (§17.5.2.34; §17.5.2.33; §17.5.2.35; §17.5.2.36) for the nearest ancestor structured document tag shall be interpreted to contain placeholder text for this structured document tag (as opposed to regular text contents within the structured document tag). If this element is present and set to `true`, this state shall be resumed (showing placeholder text) upon opening this document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then the structured document tag shall not be interpreted to be showing placeholder text when the document is displayed.

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    <w:showingPlcHdr/>
    …
    <w:richText/>
  </w:sdtPr>
  <w:sdtContent>
    <w:r>
      <w:t>[Type your name here]</w:t>
    </w:r>
 </w:sdtContent>
</w:sdt>
```

This structured document tag has run contents which read `[Type your name here]`, which would typically be interpreted as the current contents of the structured document tag. However, since the showingPlcHdr element has been specified in the structured document tag's properties, this content must instead be interpreted as the placeholder text for the structured document tag. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.5.2.40 storeMappedDataAs (Custom XML Data Date Storage Format)

This element specifies the translation which shall be performed on the displayed date in a date picker structured document tag when the current contents are saved into the associated custom XML data via the dataBinding element (§17.5.2.6).

If this element is omitted, then the value of the associated custom XML element shall be placed into the custom XML data part with no translation.

[*Example*: Consider the following date picker structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:sdt>
  <w:sdtPr>
    <w:date w:fullDate="01-01-2006T06:30:00Z">
      <w:storeMappedDateAs w:val="text"/>
      …
    </w:date>
  </w:sdtPr>
  <w:sdtContent>
    <w:r>
      <w:t>January 1</w:t>
    </w:r>
  </w:sdtContent>
</w:sdt>
```

The value of the storeMappedDateAs element's attribute value is `text`, therefore the current run contents must be sent to the mapped XML element without any translation (in this case, the value must be `January 1`). *end example*]

Attributes	Description
val (Date Storage Type)	Specifies the date translation which shall be applied to the parent date picker structured document tag. If this attribute is omitted, this its value shall be assumed to be `text`. [*Example*: Consider the following date picker structured document tag: `<w:sdt> <w:sdtPr> <w:date … > <w:storeMappedDateAs w:val="date"/> … </w:date> </w:sdtPr> <w:sdtContent> <w:r> <w:t>January 1</w:t> </w:r> </w:sdtContent> </w:sdt>` The value of the val attribute is `text`, therefore the current run contents must be sent to the mapped XML element after being translated into xsd:date format (in this case, the value must be 01-01-2006). *end example*] The possible values for this attribute are defined by the ST_SdtDateMappingType simple type (§17.18.76).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtDateMappingType) is located in §A.1. *end note*]

17.5.2.41 tabIndex (Structured Document Tag Navigation Order Index)

This element specifies the position of the current structured document tag in the navigation (tab) order used in the document. The index shall be stored on this element's val attribute and is analogous to the tabIndex attribute in HTML.

Objects that support tab index shall be navigated by consumers in the following order:

- Objects for which the XML specifies a non-zero tabIndex value are navigated first. Navigation proceeds with the element with the lowest resolved value of tabIndex to the element with the highest resolved value of tabIndex.
    - Objects that specify identical resolved values of tabIndex is navigated in the lexical order in which the elements appear in the underlying WordprocessingML.
- Objects for which the XML does not specify an index or objects for which the XML specifies a resolved tabIndex value of 0 are navigated last. These objects are navigated in the lexical order in which they appear in the underlying WordprocessingML.

[*Example*: Consider the following two structured document tags where each structured document tag specifies a tab index:

```
<w:sdt>
  <w:sdtPr>
    <w:id w:val="5" />
    <w:tabIndex w:val="1" />
  </w:sdtPr>
  <w:sdtContent>
    <w:p>
      <w:r>
        <w:t>First Name</w:t>
      </w:r>
    </w:p>
  </w:sdtContent>
</w:sdt>
…
<w:sdt>
  <w:sdtPr>
    <w:id w:val="6" />
    <w:tabIndex w:val="2" />
  </w:sdtPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <w:sdtContent>
      <w:p>
        <w:r>
          <w:t>Last Name</w:t>
        </w:r>
      </w:p>
    </w:sdtContent>
  </w:sdt>
```

The tabIndex element specifies that the structured document tag with an identifier value of 5 must be the first content to be reached via tabbing, whereas the structured document tag with an identifier value of 6 must be the second content to be reached via tabbing. *end example*]

Attributes	Description
val (Positive Decimal Number Value)	Specifies that the contents of this attribute contains a positive decimal number. The contents of this positive decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of type ST_UnsignedDecimalNumber: `<… w:val="15" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_UnsignedDecimalNumber simple type (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedDecimalNumber) is located in §A.1. *end note*]

17.5.2.42 tag (Programmatic Tag)

This element specifies a programmatic tag associated with the current structured document tag. A *programmatic tag* is an arbitrary string which applications can associate with a structured document tag in order to identify it without providing a visible friendly name. The string representing the programmatic tag shall be stored on this element's val attribute.

If this element is omitted, then no programmatic tag shall be associated with the given structured document tag.

[*Example*: Consider the following properties on a structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdtPr>
  <w:tag w:val="Clause_3246"/>
  …
</w:sdtPr>
```

This set of properties specifies via the tag element that the programmatic tag for the nearest ancestor structured document tag must be `Clause_3246`. This information can then be used as needed by applications. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.5.2.43 temporary (Remove Structured Document Tag When Contents Are Edited)

This element specifies whether the nearest ancestor structured document tag shall be removed from the WordprocessingML document when the its contents are modified.

[*Note*: This setting is primarily intended for creating structured document tags whose sole purpose is one-time placeholder text, and which should not return once replaced with content. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, then the nearest ancestor structured document tag shall not be automatically removed when its contents are modified.

[*Example*: Consider the following plain text structured document tag:

```
<w:sdt>
  <w:sdtPr>
    <w:temporary/>
    <w:text/>
  </w:sdtPr>
  …
</w:sdt>
```

This plain text structured document tag's properties contain a temporary element, specifying that the structured document tag itself must be deleted from the document whenever its contents are first modified. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.5.2.44 text (Plain Text Structured Document Tag)

This element specifies that the nearest ancestor structured document tag shall be a plain text box when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag shall satisfy the following restraints or the document shall be considered non-conformant:

- The contents shall only be contain a single run (one set of formatting properties) with exceptions for soft carriage returns via the multiLine attribute on this element
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

[*Example*: Consider the following structured document tag:

```
<w:sdt>
  <w:sdtPr>
    …
    <w:text/>
  </w:sdtPr>
  …
</w:sdt>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The text element in this structured document tag's properties specify that the kind of structured document tag is a plain text box. *end example*]

Attributes	Description

multiLine (Allow Soft Line Breaks)	Specifies whether soft line breaks can be added to the contents of this structured document tag when this document is modified. This setting shall not affect the ability of the structured document tag to display existing soft line breaks (which shall be preserved) and shall only affect the ability to add line breaks when the document is modified by an application.
	
	If this attribute is omitted, then the parent plain text structured document control shall not allow soft line breaks to be added to its contents.
	
	[*Example*: Consider the following structured document tag:
	
	`<w:sdt>`
	`  <w:sdtPr>`
	`  …`
	`  <w:text w:multiLine="true"/>`
	`  </w:sdtPr>`
	`  …`
	`</w:sdt>`
	
	The multiLine attribute on the text element in this structured document tag's properties specify that an application can allow soft line breaks to be added to the run contents of the structured document tag. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SdtText) is located in §A.1. *end note*]

17.6 Sections

WordprocessingML does not natively store the concept of pages, since it is based on paragraphs and runs (which are laid out on to pages by consumers of this content). However, although there is no concept of storing pages in the WordprocessingML format, it is often necessary to store information about a page or group of pages in a document, in order to store information that is to be used to format the pages on which a set of paragraphs appear. In WordprocessingML, this information is stored via the use of *sections*.

In WordprocessingML, *sections* are groupings of paragraphs that have a specific set of properties used to define the pages on which the text appears, as well as other section-level (applying to all paragraphs' appearance) properties.

[*Example*: Consider a document with four paragraphs of text that is to be printed on a page in landscape mode, followed by ten paragraphs of text that are to be printed in portrait mode. This requirement implies information

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

about the page(s) used to lay out each grouping of text—the first four paragraphs could require one page, or ten.

Therefore, rather than try to cache knowledge of the number of pages and their properties (which is likely to become incorrect if the XML is manipulated by a producer that does not understand page layout), this information is stored by breaking the document into two sections, as follows:

```
<w:p>
  …
</w:p>
<w:p>
  …
</w:p>
<w:p>
  …
</w:p>
<w:p>
  <w:sectPr>
    …
    (section one properties go here)
    <w:pgSz … w:orient="landscape" />
    …
  </w:sectPr>
  …
</w:p>
…
<w:p>
  <w:sectPr>
    …
    (section two properties go here)
    <w:pgSz … w:orient="landscape" />
    …
  </w:sectPr>
  …
</w:p>
```

end example]

17.6.1 bidi (Right to Left Section Layout)

This element specifies that this section shall be presented using a right-to-left page direction. This property only affects section-level properties, and does not affect the layout of text within the contents of this section.

[*Example*: Consider a section with the bidi property set as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sectPr>
  …
  <w:bidi/>
</w:sectPr>
```

This section direction is now right-to-left, which means that all section level properties are displayed right-to-left (e.g., page numbers are displayed on the right of text; columns are populated from right-to-left). However, the layout of text is determined by properties applied at the text level. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.6.2 bottom (Bottom Border)

This element specifies the presentation and display of the page border displayed at the bottom of each page in this section.

[*Example*: Consider a section in which all pages must have a bottom border consisting of a repeated image of an apple, like this:

This border would result in the following WordprocessingML:

```
<w:sectPr>
  …
  <w:pgBorders>
    <w:bottom w:val="apples" …/>
  </w:pgBorders>
  …
</w:sectPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Because the page only has a border at the bottom, only the bottom element is specified within the set of page borders. *end example*]

When a document has a bottom border that is relative to the page edges (using the offsetFrom attribute on pgBorders), it shall span the bottom edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding left or right page border (if one is specified).
- It reaches the edge of the page.

[*Example*: In the example above, no left or right border was specified in the WordprocessingML, so a consumer must draw the border from one edge of the page to the other. *end example*]

When a document has a bottom border that is relative to the text (using the offsetFrom attribute on pgBorders), it shall span only the necessary width to satisfy the requirement of spanning the width of the text.

When a document has custom border art specified by the bottomLeft, bottomRight, and/or id attributes, it shall use the corresponding relationship part item as an image for the bottom left corner, bottom right corner, and/or bottom border, respectively. If the corresponding relationship part item cannot be located, the consumer shall use the border specified by the value of the val attribute. If the corresponding value of the val attribute cannot be resolved no bottom left corner, bottom right corner, or bottom border is present when the page is displayed.

When a document has a custom border art specified by the id attribute without specifying the bottomRight and/or bottomLeft attributes, the bottom border as resolved by the corresponding relationship part item of the id attribute shall span to the corners not specified by bottomRight and/or bottomLeft attributes.

Attributes	Description
bottomLeft (Custom Defined Bottom Left Border Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which contains the custom bottom left border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom bottom left border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom left border in a document: `<w:bottom w:val="custom" r:bottomLeft="rIdCustomBottomLeftBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomLeftBorder` must contain the custom bottom left border image for the document. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
bottomRight (Custom Defined Bottom Right Border Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which contains the custom bottom right border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom bottom left border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom right border in a document: `<w:bottom w:val="custom" r:bottomRight="rIdCustomBottomRightBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomRightBorder` must contain the custom bottom right border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
color (Border Color)	Specifies the color for this border. This value can be defined as either: • A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*] . RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the border color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*] [*Example*: Consider a border color with value `auto`, as follows: `<w:bottom … w:color="auto"/>` This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
frame (Create Frame Effect)	Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text. If this attribute is omitted, then the border is not given any frame effect. [*Example*: Consider a bottom border which must appear with a frame effect, which is specified in the following WordprocessingML: `<w:bottom w:frame="true" … />` This frame's val is `true`, indicating that the border frame effect must be applied. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
id (Custom Defined Border Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
shadow (Border Shadow)	Specifies whether this border should be modified to create the appearance of a shadow.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following WordprocessingML: `<w:bottom w:shadow="true" … />` This frame's val is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

space (Border Spacing Measurement)	Specifies the spacing offset that shall be used to place this border on the parent object.
	
	When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§17.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points.
	
	When a document has a page border that is relative to the text extents (using a value of `text` in the offsetFrom attribute on pgBorders (§17.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.
	
	[*Example*: Consider a document with a set of page borders all specified to appear 24 points from the edge of the page. The resulting WordprocessingML would be as follows:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:bottom … w:space="24" />`
	`</w:pgBorders`
	
	The offsetFrom attribute specifies that the space value provides the offset of the page border from the page edge, and the value of the space attribute specifies that the page offset must be 24 points. *end example*]
	
	The possible values for this attribute are defined by the ST_PointMeasure simple type (§17.18.68).

sz (Border Width)	Specifies the width of the current border. If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<w:top w:val="dashed" w:sz="24" …/>`
	`<w:left w:val="dashed" w:sz="24" …/>`
	`<w:bottom w:val="dashed" w:sz="24" …/>`
	`<w:right w:val="dashed" w:sz="24" …/>`
	
	The border style is specified using the val attribute, and because that border style is a line border (dashed), the sz attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the ST_EighthPointMeasure simple type (§17.18.23).

themeColor (Border Theme Color)	Specifies the base theme color used to generate the border color. The border color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to be set centrally in the document.
	
	To determine the color to display, the following actions are performed:
	• Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read.
	• Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color.
	• The specified color is modified based on the presence of the themeTint or themeShade attribute.
	
	[*Example*: Consider a set of borders configured to use the accent2 theme color, resulting in the following WordprocessingML markup:
	
	`<w:top … w:themeColor="accent2" w:themeTint="99" />`
	`<w:bottom … w:themeColor="accent2" w:themeTint="99" />`
	`<w:left … w:themeColor="accent2" w:themeTint="99" />`
	`<w:right … w:themeColor="accent2" w:themeTint="99" />`
	
	If the Settings part contained the following markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:clrSchemeMapping … w:accent2="accent2"/> and the Theme part contained the following XML markup: <a:accent2> <a:srgbClr val="4F81BD"/> </a:accent2> the resulting border color would be 95B3D7 (the result of a 60% tint applied to the original theme color; see the calculations in themeTint below for details). *end example*] The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).
themeShade (Border Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border. [*Example*: Consider a shade of 40% applied to a border in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * \text{Shade}_{\text{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is C0504D.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$and converting back to RGB, we get 943634. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Border Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 \\ = 153 \\ = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Tint}_{\mathrm{pct}} + \left(1 - \mathrm{Tint}_{\mathrm{pct}}\right)$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `4F81BD`. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6)$$ $$= 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get `95B3D7`. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Border Style)	Specifies the style of border used on this object. This border can either be an art border (a repeated image along the borders - shall only be used for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<w:left w:val="single" …/>` This border's val is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_Border simple type (§17.18.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BottomPageBorder) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.6.3 col (Single Column Definition)

This element specifies the properties for a single column of text within this section.

[*Example*: Consider a single column with a width of two inches, which also has a one-inch space after the column, resulting in the following WordprocessingML:

```
<w:cols … >
  <w:col w:w="2880" w:space="1440"/>
  …
</w:cols>
```

The resulting column specifies its width of 2,880 twentieths of a point and space following of 1,440 twentieths of a point. *end example*]

The contents of the col element are only used to calculate the number and size of columns if the fixedWidth attribute is set to `false` or omitted.

Attributes	Description
space (Space Before Following Column)	Specifies the spacing (in twentieths of a point) between the current column and the next column. [*Example*: Consider a text column that is to have a one-inch space after it. This text column spacing would therefore be 1x72=144 points wide, which translates to 1,440 twentieths of a point. The resulting WordprocessingML specifies that spacing width in twentieths of a point: `<w:col … w:space="1440"/>` *end example*] For the last text column in the section, no spacing is allowed after the column, and, if present, any space value is ignored. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
w (Column Width)	Specifies the width (in twentieths of a point) of this text column. [*Example*: Consider a text column, which is to be two inches wide. This text column would therefore be 2x72=144 points wide, which translates to 2,880 twentieths of a point. The resulting WordprocessingML specifies that column width in twentieths of a point: `<w:col … w:w="2880"/>` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This attribute must be present when the parent cols element has an equalWidth attribute value of false (or equivalent). The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Column) is located in §A.1. *end note*]

17.6.4 cols (Column Definitions)

This element specifies the set of columns defined for this section in the document.

[*Example*: Consider a document in which a section defines two columns of 4.16" and 1.83", respectively, resulting in the following WordprocessingML:

```
<w:cols w:equalWidth="0">
  <w:col w:w="2640" w:space="720"/>
  <w:col w:w="6000"/>
</w:cols>
```

The cols element defines the set of columns defined for this section, which because equalWidth is `0`, are defined by the number of col elements contained in the column definition. In this case, the first column is `2,640` twentieths of a point wide (as 2640/1440ths of an inch equals 1.83 inches) with one-half of an inch space after, and the second column is `6,000` twentieths of a point wide (4.16 inches). *end example*]

Based on the presence of the equalWidth attribute, a consumer shall render the columns using:

- If equalWidth is `true`, then the columns are defined using the data stored as attributes of the cols element (defined below).
- If equalWidth is `false`, then the columns are defined using the presence and data on each child col element (§17.6.3).

Attributes	Description
equalWidth (Equal Column Widths)	Specifies whether all text columns in the current section are of equal width. If this attribute is present and its value is set to `true` or `1`, then all columns for this text section are of an equal width and are calculated as follows: • Take width of page (from margin to margin) • Divide by number of columns specified in num attribute • For each column, leave space after as defined in the space attribute • Remaining width of each column is the text column width. If this attribute is present and its value is set to `false` or `0`, then all columns for this text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	section are of different widths and are defined by each col element as follows: • Each col element defines a single column • Each w attribute defines the text column width • Each space attribute defines the space after the text column [*Example*: Consider a section with column information defined as follows: `<w:cols w:num="3" w:space="1440" w:equalWidth="1"> <w:col w:w="2880" w:space="2880" /> <w:col w:w="2880" w:space="1440" /> <w:col w:w="2880" /> </w:cols>` This set of columns has a equalWidth value set to 1, therefore the col elements are ignored, and there are three equally sized columns (num value of 3), each with one inch (space value of 1440 twentieths of a point) of space after. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
num (Number of Equal Width Columns)	Specifies the number of text columns in the current section. If all columns are not of equal width (the equalWidth attribute is not set), then this element is ignored, and the number of columns is defined by the number of col elements defined under the cols element. [*Example*: Consider a section with column information defined as follows: `<w:cols w:num="3" w:space="1440" w:equalWidth="1"> … </w:cols>` This set of columns has a equalWidth value set to 1, therefore there are three equally sized columns, as the num attribute has a value of 3. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
sep (Draw Line Between Columns)	Specifies if a vertical line is drawn between each of the text columns in this section. If set to true or 1, then a vertical line shall be drawn in the center of the spacing between each column in this section. [*Example*: Consider a section with column information defined as follows: `<w:cols w:sep="1"> …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</w:cols> This set of columns has a sep value set to 1, therefore there must be a vertical line separating each column in this section. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
space (Spacing Between Equal Width Columns)	Specifies the spacing between text columns in the current section. If all columns are not of equal width (the equalWidth attribute is not set), then this element is ignored, and the spacing after columns is defined by the space attribute on each of the col elements defined under the cols element. [*Example*: Consider a section with column information defined as follows: <w:cols w:num="3" w:space="1440" w:equalWidth="1"> … </w:cols> This set of columns has a equalWidth value set to 1, therefore there are three equally sized columns, each with one inch (space value of 1440 twentieths of a point) of space after. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

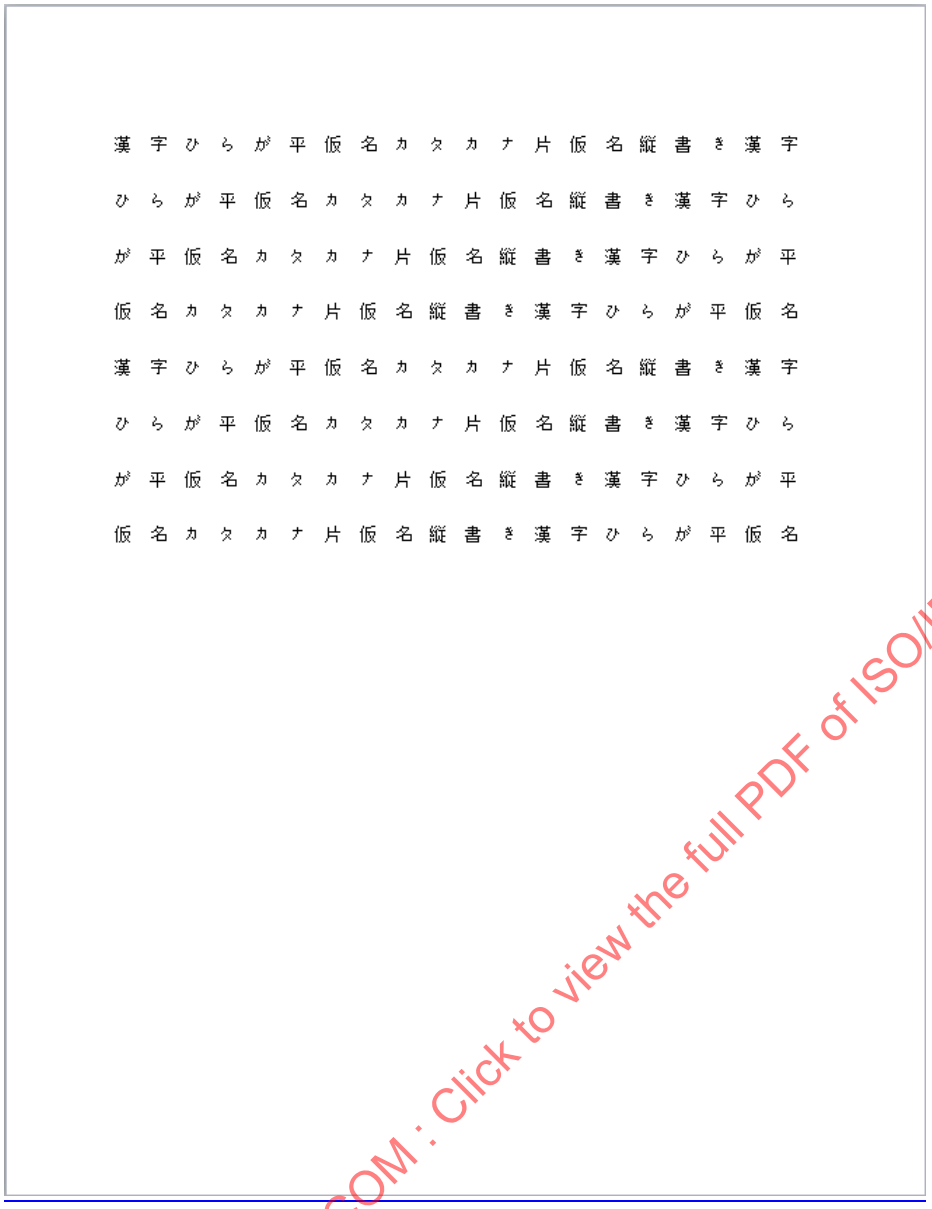
[*Note*: The W3C XML Schema definition of this element's content model (CT_Columns) is located in §A.1. *end note*]

17.6.5 docGrid (Document Grid)

This element specifies the settings for the document grid, which enables precise layout of full-width East Asian language characters within a document by specifying the desired number of characters per line and lines per page for all East Asian text content in this section.

[*Example*: Consider a document with the document grid defined to allow 20 characters per line, and 20 lines per page by snapping characters to the grid (type attribute of snapToChars) as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



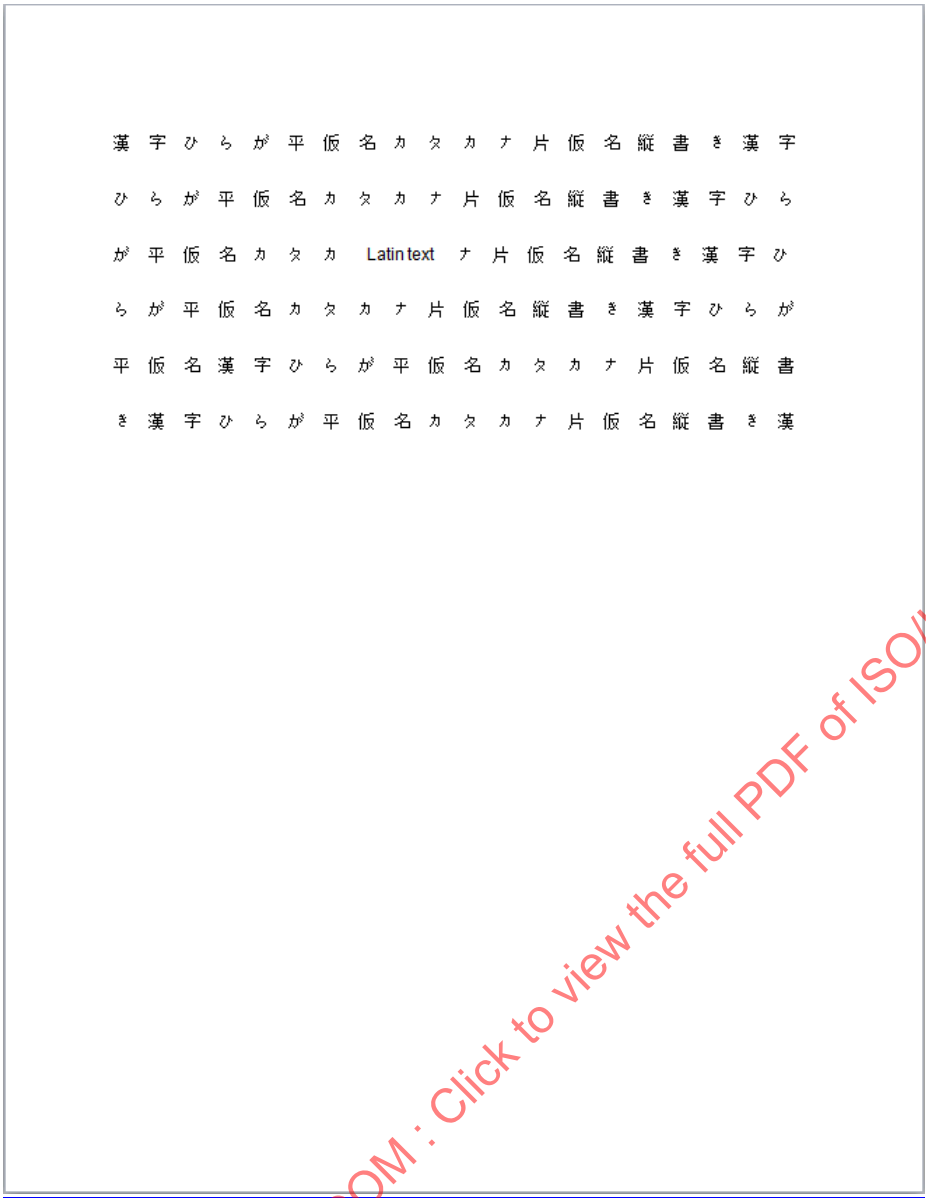
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

As shown, this document allows for only 20 East Asian characters per line by adjusting the inter-character spacing to ensure that there are only 20 characters per line. *end example*]

If Latin text is interspersed on this line, then it is placed across the number of grid units needed to fit the content, but all other grid positions are unaffected.

[*Example*: Consider the example above with the addition of the text "Latin text" in English, as follows:

 ©ISO/IEC 2016 – All rights reserved



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The Latin text spans two grid units, so it is placed in the center of those two units; no other grid positions are affected, so the text on the second line now spans two additional grid units. *end example*]

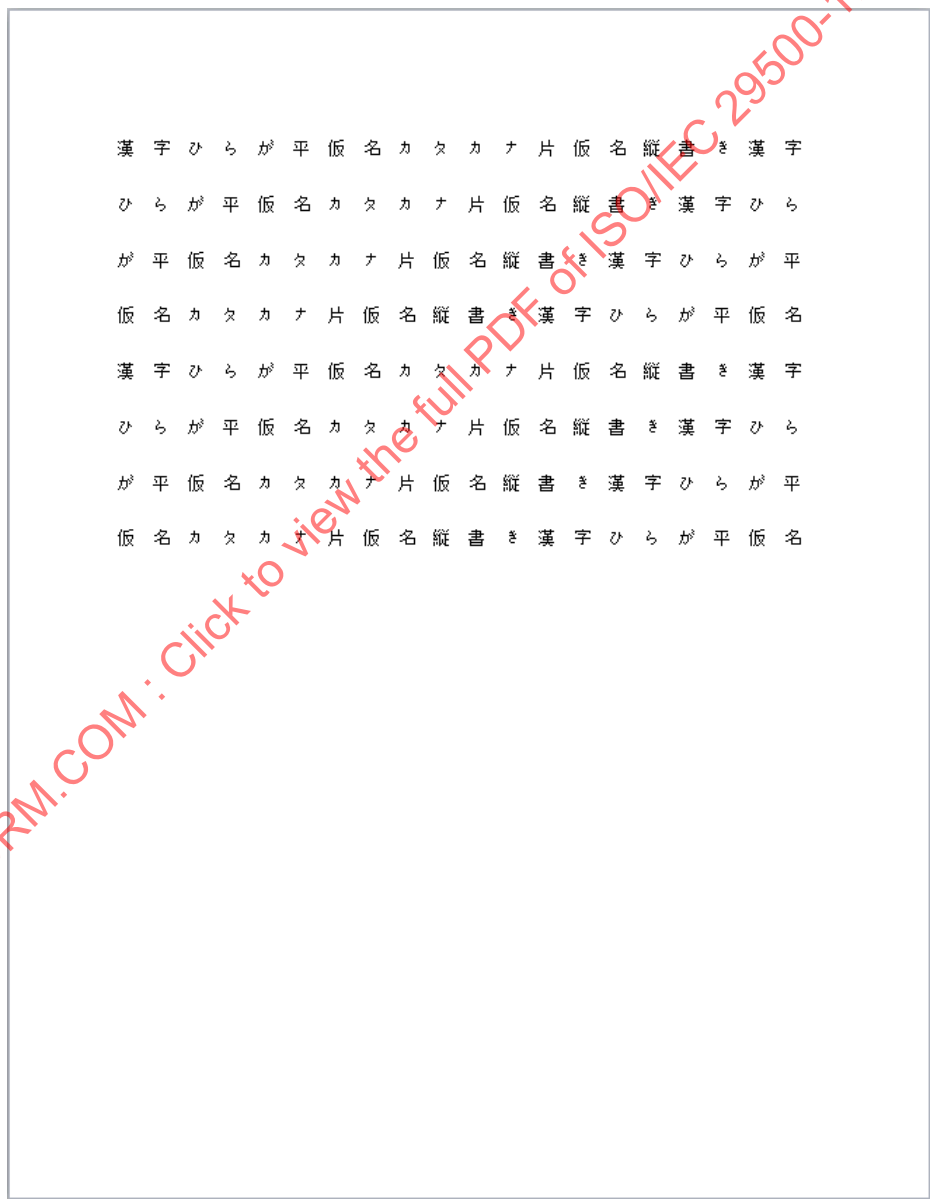
Attributes	Description
charSpace (Document Grid Character Pitch)	Specifies the number of characters to be allowed on the document grid for each line in this section. This attribute's value shall be specified by multiplying the difference between the desired character pitch and the character pitch for that character in the font size of the Normal font by 4096. If this attribute is omitted, the default value is zero. This value shall then be used to add the character pitch for the specified point size to each character in the section [Note: This results in text in the Normal style having a

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	specific number of characters per line. end note] [*Example*: Consider a section with a Normal font size of 11 points on which a 21 point pitch document grid has been defined.. The resulting WordprocessingML would be defined as follows: `<w:docGrid w:charSpace="40960" …/>` The charSpace attribute specifies a value of `40960`, which means that the delta between the character pitch of each character in the grid and the Normal font is 10 points, resulting in a character pitch of 11+10 = 21 points for all characters in this section. *end example*] Individual runs of text can override the line pitch information specified for the document grid by specifying that the run text shall not snap to the document grid via the snapToGrid element (§17.3.2.34). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
linePitch (Document Grid Line Pitch)	Specifies the number of lines to be allowed on the document grid for the current page assuming all lines have equal line pitch applied to them. This line pitch shall not be added to any line which appears within a table cell unless the adjustLineHeightInTable element (§17.15.3.1) is present in the document's compatibility settings. This attribute is specified in twentieths of a point, and defines the pitch for each line of text on this page such that the desired number of single spaced lines of text fits on the current page. [*Example*: Consider a standard 8.5x11" page on which a 20 character wide, 20 line document grid has been defined. The resulting WordprocessingML would be defined as follows: `<w:docGrid w:linePitch="684" …/>` The linePitch attribute specifies that 34.2 points is to the amount of pitch allowed for each line on this page in order to maintain the specific document grid. *end example*] Individual paragraphs can override the line pitch information specified for the document grid by either: • Specifying an exact line spacing value using the lineRule attribute of value `exact` on the spacing element (§17.3.1.33). • Specifying that the paragraph text shall not snap to the document grid via the snapToGrid element (§17.3.1.32). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
type (Document Grid Type)	Specifies the style of the current document grid, which defines the grid behavior. The grid can define a grid which snaps all East Asian characters to grid positions, but leaves Latin text with its default spacing; a grid which adds the specified character pitch to each character on each row; or a grid which affects only the line pitch for the current section. [*Example*: Consider the document discussed above with the document grid defined to allow 20 characters per line, and 20 lines per page by snapping characters to the grid as follows: 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	This document has a type attribute of snapToChars, which specifies that the grid must force East Asian characters to fit 20 to a line. *end example*] If this attribute is omitted, the value "default" shall be used. [*Note:* Disabling the document grid in this way rather than omitting the docGrid element allows for the preservation of document grid settings if and when the document grid is re-enabled. *end note*] The possible values for this attribute are defined by the ST_DocGrid simple type (§17.18.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocGrid) is located in §A.1. *end note*]

17.6.6 formProt (Only Allow Editing of Form Fields)

This element specifies that the contents of the current section shall be protected such that they cannot be edited by a user (if the consumer is displaying the document and allowing the user to make modification) except for the text contained in any form field or embedded control that is part of the current section.

[*Example*: Consider a section consisting of three paragraphs of text and a single text form field, located at the beginning of the second paragraph. If this section is protected in this manner, a user would only be permitted to edit the contents of the text form field, and all other contents would be locked to prevent user edits. *end example*]

The enforcement of this property is determined by the documentProtection element (§17.15.1.29), as it is possible to specify protection without turning it on.

This element's content model is defined by the common boolean property definition in §17.17.4.

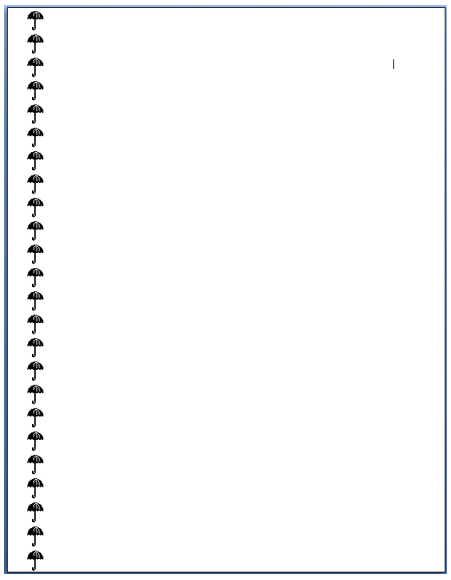
17.6.7 left (Left Border)

This element specifies the presentation and display of the page border displayed at the left of each page in this section.

[*Example*: Consider a section in which all pages have a left border consisting of a repeated image of an umbrella, like this:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



This border would result in the following WordprocessingML:

```
<w:sectPr>
  …
  <w:pgBorders>
    <w:left w:val="seattle" …/>
  </w:pgBorders>
  …
</w:sectPr>
```

Because the page only has a border at the left, only the left element is specified within the set of page borders. *end example*]

When a document has a left border that is relative to the page edges (using the offsetFrom attribute value of page on pgBorders), it shall span the left edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding top or bottom page border (if one is specified).
- It reaches the edge of the page.

[*Example*: In the example above, no top or bottom border was specified in the WordprocessingML, so a consumer must draw the border from one edge of the page to the other. *end example*]

When a document has a left border that is relative to the text (using the offsetFrom attribute value of `text` on pgBorders), it shall span only the necessary width to satisfy the requirement of spanning the width of the text.

When a document has custom border art specified by the id attribute, it shall use the corresponding relationship part item as an image for the left border. If the corresponding relationship part item cannot be located, the consumer shall use the border specified by the value of the val attribute. If the corresponding value of the val attribute cannot be resolved, no left border is present when the page is displayed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
color (Border Color)	Specifies the color for this border. This value can be defined as either: • A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*] . RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the border color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*] [*Example*: Consider a border color with value `auto`, as follows: `<w:bottom … w:color="auto"/>` This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*] If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
frame (Create Frame Effect)	Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text. If this attribute is omitted, then the border is not given any frame effect. [*Example*: Consider a bottom border which must appear with a frame effect, which is specified in the following WordprocessingML: `<w:bottom w:frame="true" … />` This frame's val is `true`, indicating that the border frame effect must be applied. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
id (Custom Defined Border Relationship Reference)	Specifies the relationship ID for the relationship which contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
shadow (Border Shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following WordprocessingML: `<w:bottom w:shadow="true" … />` This frame's val is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
space (Border Spacing Measurement)	Specifies the spacing offset that shall be used to place this border on the parent object. When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§17.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points. When a document has a page border that is relative to the text extents (using a value of `text` in the offsetFrom attribute on pgBorders (§17.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a document with a set of page borders all specified to appear 24 points from the edge of the page. The resulting WordprocessingML would be as follows:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:bottom … w:space="24" />`
	`</w:pgBorders`
	
	The offsetFrom attribute specifies that the space value provides the offset of the page border from the page edge, and the value of the space attribute specifies that the page offset must be 24 points. *end example*]
	
	The possible values for this attribute are defined by the ST_PointMeasure simple type (§17.18.68).

sz (Border Width)	Specifies the width of the current border.
	
	If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<w:top w:val="dashed" w:sz="24" …/>`
	`<w:left w:val="dashed" w:sz="24" …/>`
	`<w:bottom w:val="dashed" w:sz="24" …/>`
	`<w:right w:val="dashed" w:sz="24" …/>`
	
	The border style is specified using the val attribute, and because that border style is a line border (dashed), the sz attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the ST_EighthPointMeasure simple type (§17.18.23).

themeColor (Border Theme Color)	Specifies the base theme color used to generate the border color. The border color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor. The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	be set centrally in the document.
	
	To determine the color to display, the following actions are performed:
	• Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read.
	• Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color.
	• The specified color is modified based on the presence of the themeTint or themeShade attribute.
	
	[*Example*: Consider a set of borders configured to use the accent2 theme color, resulting in the following WordprocessingML markup:
	
	`<w:top … w:themeColor="accent2" w:themeTint="99" />`
	`<w:bottom … w:themeColor="accent2" w:themeTint="99" />`
	`<w:left … w:themeColor="accent2" w:themeTint="99" />`
	`<w:right … w:themeColor="accent2" w:themeTint="99" />`
	
	If the Settings part contained the following markup:
	
	`<w:clrSchemeMapping … w:accent2="accent2"/>`
	
	and the Theme part contained the following XML markup:
	
	`<a:accent2>`
	`  <a:srgbClr val="4F81BD"/>`
	`</a:accent2>`
	
	the resulting border color would be 95B3D7 (the result of a 60% tint applied to the original theme color; see the calculations in themeTint below for details). *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Border Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border. [*Example*: Consider a shade of 40% applied to a border in a document. This shade is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	calculated as follows: $$S_{xml} = 0.4 * 255 \\ = 102 \\ = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Shade}_{\mathrm{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is C0504D. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get 943634. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Border Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 \\ = 153 \\ = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Tint}_{\mathrm{pct}} + (1 - \mathrm{Tint}_{\mathrm{pct}})$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `4F81BD`. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6) \\ = 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get `95B3D7`. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Border Style)	Specifies the style of border used on this object. This border can either be an art border (a repeated image along the borders - shall only be used for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<w:left w:val="single" …/>` This border's val is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_Border simple type (§17.18.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageBorder) is located in §A.1. *end note*]

17.6.8 lnNumType (Line Numbering Settings)

This element specifies the settings for line numbering to be displayed before each column of text in this section in the document.

[*Example*: Consider the line numbering used on each page of this document, which specifies: line numbering for each line, restarting at one at the top of each new page. This line-numbering scheme would be defined using the following WordprocessingML:

```
<w:sectPr>
  …
  <w:lnNumType w:countBy="1" />
</w:sectPr>
```

This content specifies that line numbers shall be included on each line, restart on each page (the default), be placed automatically based on the text (the default), and shall restart at one (the default). *end example*]

Attributes	Description
countBy (Line Number Increments to Display)	Specifies the line number increments to be displayed in the current document. Although each line has an associated line number, only lines which are an even multiple of this value shall be displayed. If this attribute is missing, no line numbering shall be applied to the section. [*Example*: Consider a document in which only every fifth line must have a line number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The resulting WordprocessingML for this setting would be: `<w:lnNumType … w:countBy="5"/>` This setting ensures that only lines whose number is a multiple of (e.g. 5, 10, and 15) has a line number displayed. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
distance (Distance Between Text and Line Numbering)	Specifies the distance between the text margin and the edge of any line numbers appearing in that section. [*Example*: Consider a document in which the line numbering must appear one-half inch from the text margin. The WordprocessingML for this setting is: `<w:lnNumType … w:distance="720"/>` The distance attribute specifies that there must be a 720 twip spacing between the text margin and the *line numbering. end example*] If this attribute is missing, the line number positioning according to the surrounding text is implementation-defined. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

restart (Line Numbering Restart Setting)	Specifies when the line numbering in this section shall be reset to the line number specified by the start attribute's value.
	
	The line numbering increments for each line (even if it is not displayed) until it reaches the restart point specified by this element.
	
	[*Example*: Consider the line numbering used on each page of this document, which specifies that line numbering must restart at the top of each new page. This line numbering setting would be defined using the following WordprocessingML:
	
	`<w:sectPr>`
	`…`
	`<w:lnNumType w:restart="newPage" … />`
	`</w:sectPr>`
	
	The value of newPage specifies that the line numbers must restart at the top of each page to the value specified by the start attribute. In this case, newPage is the default, so this value could have been omitted entirely. *end example*]
	
	The possible values for this attribute are defined by the ST_LineNumberRestart simple type (§17.18.47).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
start (Line Numbering Starting Value)	Specifies the starting value used for the first line whenever the line numbering is restarted by use of the restart attribute. [*Example*: Consider a document in which line numbering must appear on every fifth line, but the first line must be treated as line number . This setting would require the following WordprocessingML syntax: `<w:lnNumType w:start="3" w:countBy="5"/>` The start attribute specifies that line numbers must be counted starting from the number 3. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineNumber) is located in §A.1. *end note*]

17.6.9 paperSrc (Paper Source Information)

This element specifies printer-specific settings for the printer tray(s) that shall be used to print different pages in this section in the document.

[Example: Consider a section which must use the best possible tray when printing all pages in this section. This information is specified using the following WordprocessingML:

```
<w:paperSrc w:first="1" w:other="1" />
```

The attributes on the paperSrc element specify the printer codes for the trays to be used when printing this section. end example]

Attributes	Description
first (First Page Printer Tray Code)	Specifies a printer-specific code that uniquely identifies a specific printer tray to be used to print the first page of this section in the document. A first value of 1 (the default) is specifically used to indicate that the printer shall automatically select the appropriate printer tray based on the printed page size. [*Example*: Consider a section which must use the best possible tray when printing the first page in this section. This information is specified using the following WordprocessingML: `<w:paperSrc w:first="1" w:other="1" />` The first attribute on the paperSrc element specifies that the printer must automatically

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	select the appropriate printer tray based on the printed page size when printing the first page in this section. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
other (Non-First Page Printer Tray Code)	Specifies a printer-specific code that uniquely identifies a specific printer tray to be used to print the each subsequent (non-first) page of this section in the document. An value of 1 (the default) is specifically used to indicate that the printer shall automatically select the appropriate printer tray based on the printed page size. [*Example*: Consider a section which must use the best possible tray when printing the all pages in this section. This information is specified using the following WordprocessingML: `<w:paperSrc w:first="1" w:other="1" />` The other attribute on the paperSrc element specifies that the printer must automatically select the appropriate printer tray based on the printed page size when printing all pages after the first in this section. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

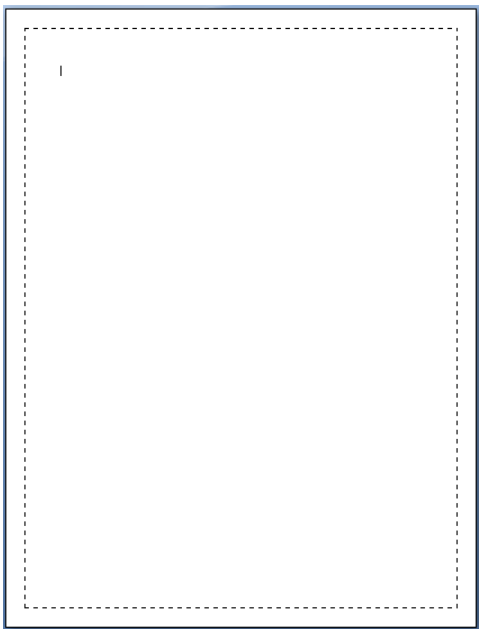
[*Note*: The W3C XML Schema definition of this element's content model (CT_PaperSource) is located in §A.1. *end note*]

17.6.10 pgBorders (Page Borders)

This element specifies the page borders for each page in this section. Each child element of the pgBorders element specifies a specific of border (left, right, bottom, or top).

[*Example*: Consider a page that specifies a dashed line border around each of the four sides of the page, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



This page border setting would be specified using the following WordprocessingML:

```
<w:pgBorders w:offsetFrom="page">
  <w:top w:val="dashed" w:sz="4" w:space="24" w:color="auto" />
  <w:left w:val="dashed" w:sz="4" w:space="24" w:color="auto" />
  <w:bottom w:val="dashed" w:sz="4" w:space="24" w:color="auto" />
  <w:right w:val="dashed" w:sz="4" w:space="24" w:color="auto" />
</w:pgBorders>
```

The four page borders are each uniquely defined by the top, left, bottom, and right elements, respectively. Global settings that define the placement of all page borders are stored on the pgBorders element directly. *end example*]

Attributes	Description

display (Pages to Display Page Borders)	Specifies the pages in this section on which the page border shall be printed.
	
	If this attribute is omitted, then the page borders shall be displayed on all pages in this section (equivalent to a value of `allPages`).
	
	[*Example*: Consider a section in a document for which the page border must only be printed on the first page. This setting is specified using the following WordprocessingML:
	
	`<w:pgBorders w:display="firstPage">`
	`…`
	`</w:pgBorders>`
	
	The display attribute specifies that only the first page must display the page border defined for this section. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

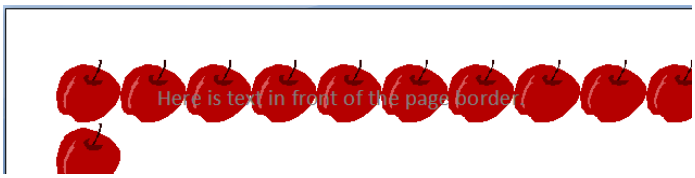
©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_PageBorderDisplay simple type (§17.18.62).

offsetFrom (Page Border Positioning)	Specifies how the relative positioning of the page borders shall be calculated.
	
	If the value of this attribute is page, then the space attribute on each page border shall be interpreted as the distance from the edge of the page that shall be left before the page border.
	
	If the value of this attribute is text, then the space attribute on each page border shall be interpreted as the distance from the text margins that shall be left before the page border.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:top w:val="dashed" w:space="24" />`
	`  <w:left w:val="dashed" w:space="24" />`
	`  <w:bottom w:val="dashed" w:space="24"/>`
	`  <w:right w:val="dashed" w:space="24"/>`
	`</w:pgBorders>`
	
	This fragment specifies that the page borders must be indented 24 points from the page extents.
	
	This is distinct from the following fragment with identical space attribute values:
	
	`<w:pgBorders w:offsetFrom="text">`
	`  <w:top w:val="dashed" w:space="24" />`
	`  <w:left w:val="dashed" w:space="24" />`
	`  <w:bottom w:val="dashed" w:space="24"/>`
	`  <w:right w:val="dashed" w:space="24"/>`
	`</w:pgBorders>`
	
	In this case, the page borders is offset by 24 points, but in this case, that offset is calculated relative to the text margins. *end example*]
	
	The possible values for this attribute are defined by the ST_PageBorderOffset simple type (§17.18.63).

zOrder (Z-Ordering of Page Border)	Specifies whether the page border is positioned above or below intersecting texts and objects in this document. [*Example*: Consider a document in which the page border must be displayed below any intersecting text as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	 This setting is specified by setting the zOrder attribute to back, which specifies that the page border must be display behind all intersecting text and objects. *end example*] The possible values for this attribute are defined by the ST_PageBorderZOrder simple type (§17.18.64).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageBorders) is located in §A.1. *end note*]

17.6.11 pgMar (Page Margins)

This element specifies the page margins for all pages in this section.

[*Example*: Consider a page with a one-inch margin on all sides. Specifying these margins requires the following WordprocessingML:

```
<w:sectPr>
  <w:pgMar w:header="720" w:bottom="1440" w:top="1440" w:right="1440"
w:left="1440" w:footer="720" w:gutter="0" _/>
  …
</w:sectPr>
```

This section specifies page margins of 1,440 twentieths of a point (one inch) on all sides. *end example*]

[*Note*: With reference to Fig. 48 of *Requirements for Japanese Text Layout*, typical positions (a), (b), (c), (d), and (f), but not (e), of running heads and page numbers for vertically set books with double running heads, and typical positions (a), (b), (c), and (d) of running heads and page numbers for horizontally set books with double running heads, can be represented using WordprocessingML. *end note*]

Attributes	Description
bottom (Page Bottom Spacing)	Specifies the distance (in twentieths of a point) between the bottom of the text margins for the main document and the bottom of the page for all pages in this section. If the value of bottom is non-negative, then the text is placed at the greater of: • The value of bottom • The extent of the footer text [*Example*: Consider a document where the footer must start one inch of the bottom of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	the page extent, but the contents of the main document story specify that they must start one-half of an inchh from the page extents. To specify these boundaries, the following page margins can specified in the WordprocessingML:
	
	`<w:pgMar … w:bottom="720"`
	`  w:footer="1440"/>`
	
	This fragment specifies that the footer must start 1440 twentieths of a point from the bottom of the page, but the main document story must start 720 twentieths of a point from the bottom of the page. Since the footer extent is guaranteed to be greater in this case, the bottom text extent ends at the top of the footer region. *end example*]
	
	A negative value indicates that the contents of the main document shall be measured from the bottom of the page extent regardless of the footer for that document, and therefore shall overlap the footer text.
	
	[*Example*: Consider a document where the footer must start one inch of the bottom of the page extent, but the contents of the main document story must start one-half of an inch from the page extents. To specify these boundaries, the following page margins can be specified in the WordprocessingML:
	
	`<w:pgMar … w:bottom="-720"`
	`  w:footer="1440"/>`
	
	This fragment specifies that the footer must start 1440 twentieths of a point from the bottom of the page, and the main document story must start 720 twentieths of a point from the bottom of the page. Since the value of bottom is negative in this case, the bottom text extent starts one-half of an inch from the bottom of the page and overlaps any footer text. *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

footer (Spacing to Bottom of Footer)	Specifies the distance (in twentieths of a point) from the bottom edge of the page to the bottom edge of the footer. [*Example*: Consider a document where the footer must start one inch above the bottom of the page extent. To specify this boundary, the following page margins must specified in the WordprocessingML: `<w:pgMar … w:footer="1440"/>` This fragment specifies that the footer must start 1440 twentieths of a point from the bottom of the page. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
gutter (Page Gutter Spacing)	Specifies the page gutter for each page in the current section. The *page gutter* defines the amount of extra space added to the specified margin, above any existing margin values. [*Note*: This setting is typically used when a document is being created for binding, in order to ensure that the resulting margins are present after the binding gutter is consumed by the printed matter binding. *end note*] [*Example*: Consider a document where the margin must start one inch of the left edge of the page extent after one-half of an inch is hidden by the page binding. To specify this condition, a user could simply use a left margin of 1.5 inches, which would be lost if the margins are later changed, or could specify a one-half inch gutters follows in the WordprocessingML: `<w:pgMar … w:gutter="720"/>` This fragment specifies that the gutter must span 720 twentieths of a point, after which any margin value must be added. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
header (Spacing to Top of Header)	Specifies the distance (in twentieths of a point) from the top edge of the page to the top edge of the header. [*Example*: Consider a document where the header must start two inches below the top of the page extent. To specify this boundary, the following page margins must specified in the WordprocessingML: `<w:pgMar … w:header="2880"/>` This fragment specifies that the header must start 2880 twentieths of a point from the top of the page. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
left (Left Margin Spacing)	Specifies the distance (in twentieths of a point) between the left edge of the page and the left edge of the text extents for this document. [*Example*: Consider a document where the left text extent must start two inches inside the page extent. To specify this boundary, the following page margins must specified in the WordprocessingML: `<w:pgMar … w:left="2880"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This fragment specifies that the left margin must span 2880 twentieths of a point from the left edge of the page. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
right (Right Margin Spacing)	Specifies the distance (in twentieths of a point) between the right edge of the page and the right edge of the text extents for this document. [*Example*: Consider a document where the right text extent must start one inch inside the page. To specify this boundary, the following page margins must specified in the WordprocessingML: `<w:pgMar … w:right="1440"/>` This fragment specifies that the right margin must span 1440 twentieths of a point from the right edge of the page. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
top (Top Margin Spacing)	Specifies the distance (in twentieths of a point) between the top of the text margins for the main document and the top of the page for all pages in this section. If the value of top is non-negative, then the text is placed at the greater of: • The value of top • The extent of the header text [*Example*: Consider a document where the header must start one inch from the top of the page extent, but the contents of the main document story specify that they must start one-half of an inch from the page extents. To specify these boundaries, the following page margins can be specified in the WordprocessingML: `<w:pgMar … w:top="720" w:header="1440"/>` This fragment specifies that the header must start 1440 twentieths of a point from the top of the page, but the main document story must start 720 twentieths of a point from the top of the page. Since the header extent is guaranteed to be greater in this case, the main text extent ends at the bottom of the header region. *end example*] A negative value indicates that the contents of the main document shall be measured from the top of the page extent regardless of the header for that document, and therefore shall overlap the header text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a document where the header must start one inch from the top of the page extent, but the contents of the main document story must start one-half of an inch from the page extents. To specify these boundaries, the following page margins can be specified in the WordprocessingML:
	
	`<w:pgMar … w:top="-720"`
	`  w:header="1440"/>`
	
	This fragment specifies that the header must start 1440 twentieths of a point from the top of the page, and the main document story must start 720 twentieths of a point from the top of the page. Since the value of top is negative in this case, the top text extent starts one-half of an inch from the top of the page and overlaps any header text. *end example*]
	
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageMar) is located in §A.1. *end note*]

17.6.12 pgNumType (Page Numbering Settings)

This element specifies the page numbering settings for all page numbers that appear in the contents of the current section.

[*Example*: Consider a section in which the page numbers must start at page 25. The following WordprocessingML syntax specifies that requirement:

```
<w:sectPr>
  …
  <w:pgNumType w:start="25"/>
</w:sectPr>
```

The pgNumType element specifies that numbering on this section must start from page number 25. *end example*]

Attributes	Description
chapSep (Chapter Separator Character)	Specifies the separator character that shall appear between the chapter and page number, if a chapter style has been set for page numbers in this section. If the chapStyle attribute is not present, or its specified heading level does not have an associated numbering format, then this value is ignored, since no chapter number is output by the field. [Example: Consider a section in a document in which the chapter must be separated from

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	the page number using a colon character. This constraint would be specified using the following WordprocessingML: `<w:pgNumType w:chapSep="colon" w:chapStyle="1" />` The chapSep attribute declares that the chapter and page number must be separated by a colon (e.g. 1:1 for chapter one, page one). end example] The possible values for this attribute are defined by the ST_ChapterSep simple type (§17.18.6).
chapStyle (Chapter Heading Style)	Specifies the one-based index of the heading style applied to chapter titles in the document which shall be used as chapter headings in all page numbers for this section, by locating the nearest heading of that style and extracting the numbering information. If the specified heading style does not exist in the current section, or does not have a numbering format, then any previous level heading format shall be used as needed as the specified chapter number. If no heading has numbering information and/or is used in the section, then the chapter and chapter separator shall be omitted from the page numbering data. [*Example*: Consider a page number in a section with page numbering properties that specify a chapStyle of 1 (Heading 1 style) and a chapSep of dash. This means that for each page number in this section, the numbering value of the nearest Heading 1 style is used for the chapter value, and is followed by a dash, then the page n umber in that section. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
fmt (Page Number Format)	Specifies the number format that shall be used for all page numbering in this section. [*Example*: A fmt value of `lowerLetter` indicates that a consumer must use lowercase letters for each page in this section: a,b,c… *end example*] The possible values for this attribute are defined by the ST_NumberFormat simple type (§17.18.59).
start (Starting Page Number)	Specifies the page number that appears on the first page of the section. If this value is omitted, numbering continues from the highest page number in the previous section. [*Example*: Consider the following WordprocessingML: `<w:pgNumType w:fmt="lowerLetter"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	Because the start value is omitted, the page numbers in this section begin at the value of the highest page in the previous section. This means that if the previous section ended in page 7, this section would start with page 8. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageNumber) is located in §A.1. *end note*]

17.6.13 pgSz (Page Size)

This element specifies the properties (size and orientation) for all pages in the current section.

[*Example*: Consider a section that must be printed on A4 paper. The WordprocessingML for this paper size is as follows:

```
<w:pgSz w:w="11907" w:h="16839" />
```

This output states that all pages in this section must be 11907 twentieths of a point wide (11907 twentieths of a point = 8.269") and 16839 twentieths of a point high (16939 twentieths of a point = 11.694"). *end example*]

Attributes	Description
code (Printer Paper Code)	Specifies an optional value which can be used to store an identifier for the current paper size. This code is stored solely to show a description for the current paper size. This setting should not be used to determine the target paper size (i.e. if the w and h attributes are omitted, this setting has no meaning). [Example: Consider the following WordprocessingML fragment: `<w:pgSz w:w="12240" w:h="15840" w:code="240" />` The code attribute specifies a value of 240, whichis, since the w and h attributes specify a page size of 8.5 inches by 11 inches, can be used to determine the appropriate user label for this paper size – for example, "Letter" or "8.5 by 11". The attribute code specifies that the producing application's unique identifier value for this paper size was 240. This value does not itself determine the paper size, regardless of the presence of the w and h attributes. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.10).
h (Page Height)	Specifies the height (in twentieths of a point) for all pages in the current section. [*Example*: Consider the following WordprocessingML: `<w:pgSz w:w="15840" w:h="12240" />` All pages in this section are displayed on a page that is `12240` twentieths of a point (8.5") tall. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).
orient (Page Orientation)	Specifies the orientation of all pages in this section. This information is used to determine the actual paper size to use on the printer. [*Example*: Pages 11" wide by 8.5" long in landscape mode use 8.5"x11" paper, because the width and height are reversed for pages in this landscape section with respect to the printed page. *end example*] This implies that the actual paper size width and height are reversed for pages in this section. If this attribute is omitted, then `portrait` shall be implied. [*Example*: Consider the following WordprocessingML: `<w:pgSz w:w="15840" w:h="12240" w:orient="landscape" />` Although the page width is 11", and page height is 8.5", according to the w and h attributes, because the orient attribute is set to `landscape`, pages in this section are printed on 8.5x11" paper in landscape mode. *end example*] The possible values for this attribute are defined by the ST_PageOrientation simple type (§17.18.65).
w (Page Width)	This attribute indicates the width (in twentieths of a point) for all pages in the current section. [*Example*: Consider the following WordprocessingML: `<w:pgSz w:w="15840" w:h="12240" />` All pages in this section are displayed on a page that is `15840` twentieths of a point (11") wide. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageSz) is located in §A.1. *end note*]

17.6.14 printerSettings (Reference to Printer Settings Data)

This element specifies an explicit relationship to a Printer Settings part containing information about the printer settings used for this section.

If this element is omitted, than no additional settings are associated with this section.

[*Example*: Consider a producer which needed to store additional printer settings for each section. A document from such a producer would have the following section properties:

```
<w:sectPr>
  …
  <w:printerSettings r:id="rId10" />
</w:sectPr>
```

The resulting Main Document part would a relationship to the appropriate Printer Settings part with a relationship ID of `rId10`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

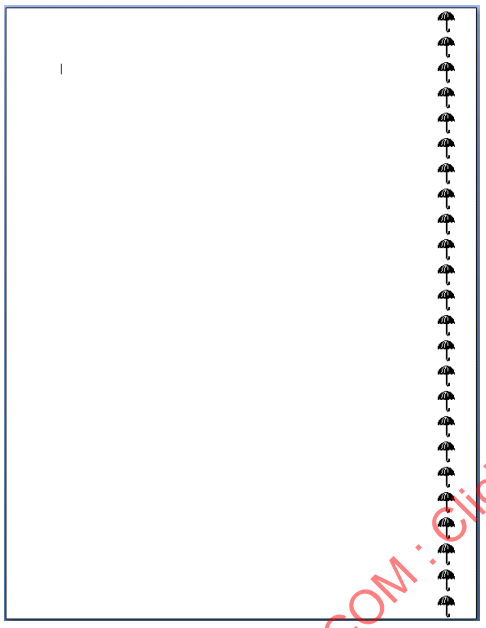
Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.6.15 right (Right Border)

This element specifies the presentation and display of the page border displayed at the right of each page in this section.

[*Example*: Consider a section in which all pages must have a right border consisting of a repeated image of an umbrella, like this:



This border would result in the following WordprocessingML:

```
<w:sectPr>
  …
  <w:pgBorders>
    <w:right w:val="seattle" …/>
  </w:pgBorders>
  …
</w:sectPr>
```

Because the page has a border at the right only, only the right element is specified within the set of page borders. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

When a document has a right border that is relative to the page edges (using the offsetFrom attribute value of page on pgBorders), it shall span the right edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding top or bottom page border (if one is specified)
- It reaches the edge of the page.

[*Example*: In the example above, no top or bottom border was specified in the WordprocessingML, so a consumer must draw the border from one edge of the page to the other. *end example*]

When a document has a right border that is relative to the text (using the offsetFrom attribute value of `text` on pgBorders), it shall only span the necessary width to satisfy the requirement of spanning the width of the text.

When a document has custom border art specified by the id attribute, it shall use the corresponding relationship part item as an image for the right border. If the corresponding relationship part item cannot be located, the consumer shall use the border specified by the value of the val attribute. If the corresponding value of the val attribute cannot be resolved, no right border is present when the page is displayed.

Attributes	Description
color (Border Color)	Specifies the color for this border. This value can be defined as either: • A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*] . RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the border color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*] [*Example*: Consider a border color with value `auto`, as follows: `<w:bottom … w:color="auto"/>` This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*] If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
frame (Create Frame Effect)	Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text. If this attribute is omitted, then the border is not given any frame effect. [*Example*: Consider a bottom border which must appear with a frame effect, which is specified in the following WordprocessingML: `<w:bottom w:frame="true" … />` This frame's val is `true`, indicating that the border frame effect must be applied. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
id (Custom Defined Border Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
shadow (Border Shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	the following WordprocessingML: `<w:bottom w:shadow="true" … />` This frame's val is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

space (Border Spacing Measurement)	Specifies the spacing offset that shall be used to place this border on the parent object.
	
	When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§17.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points.
	
	When a document has a page border that is relative to the text extents (using a value of `text` in the offsetFrom attribute on pgBorders (§17.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.
	
	[*Example*: Consider a document with a set of page borders all specified to appear 24 points from the edge of the page. The resulting WordprocessingML would be as follows:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:bottom … w:space="24" />`
	`</w:pgBorders`
	
	The offsetFrom attribute specifies that the space value provides the offset of the page border from the page edge, and the value of the space attribute specifies that the page offset must be 24 points. *end example*]
	
	The possible values for this attribute are defined by the ST_PointMeasure simple type (§17.18.68).

sz (Border Width)	Specifies the width of the current border. If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value. If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range can be reassigned to a more appropriate value. [*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:top w:val="dashed" w:sz="24" …/> <w:left w:val="dashed" w:sz="24" …/> <w:bottom w:val="dashed" w:sz="24" …/> <w:right w:val="dashed" w:sz="24" …/> The border style is specified using the val attribute, and because that border style is a line border (dashed), the sz attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*] The possible values for this attribute are defined by the ST_EighthPointMeasure simple type (§17.18.23).
themeColor (Border Theme Color)	Specifies the base theme color used to generate the border color. The border color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor. The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to be set centrally in the document. To determine the color to display, the following actions are performed: • Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read. • Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color. • The specified color is modified based on the presence of the themeTint or themeShade attribute. [*Example*: Consider a set of borders configured to use the accent2 theme color, resulting in the following WordprocessingML markup: <w:top … w:themeColor="accent2" w:themeTint="99" /> <w:bottom … w:themeColor="accent2" w:themeTint="99" /> <w:left … w:themeColor="accent2" w:themeTint="99" /> <w:right … w:themeColor="accent2" w:themeTint="99" /> If the Settings part contained the following markup: <w:clrSchemeMapping … w:accent2="accent2"/> and the Theme part contained the following XML markup: <a:accent2> <a:srgbClr val="4F81BD"/>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	`</a:accent2>` the resulting border color would be `95B3D7` (the result of a 60% tint applied to the original theme color; see the calculations in themeTint below for details). *end example*] The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).
themeShade (Border Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border. [*Example*: Consider a shade of 40% applied to a border in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeShade value in the file format would be `66`. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * Shade_{percentage}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 = 0.39698$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get 943634. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Border Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 = 153 = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * \text{Tint}_{pct} + (1 - \text{Tint}_{pct})$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is 4F81BD. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6)$$ $$= 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get 95B3D7. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/>` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
val (Border Style)	Specifies the style of border used on this object. This border can either be an art border (a repeated image along the borders - shall only be used for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<w:left w:val="single" …/>` This border's val is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_Border simple type (§17.18.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageBorder) is located in §A.1. *end note*]

17.6.16 rtlGutter (Gutter on Right Side of Page)

This element specifies that the page gutter shall be placed on the right side of the page for this section only. The *page gutter* defines the amount of extra space added to the specified margin, above any existing margin values. [*Note*: This setting is typically used when a document is being created for binding, in order to ensure that the resulting margins are present after the binding gutter is consumed by the printed matter binding. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If the gutter is set to the side of the page by the omission of the gutterAtTop element (§17.15.1.50), then each section's gutter is placed at the left by default, unless that default is overridden by the rtlGutter element.

[*Example*: Consider a document with three sections, with gutter properties defined as follows:

```
<w:p>
  <w:pPr>
    <w:sectPr>
      <w:pgMar w:gutter="1440" …/>
      …
    </w:sectPr>
  </w:pPr>
</w:p>
…
<w:p>
  <w:pPr>
    <w:sectPr>
      <w:pgMar w:gutter="1440" …/>
      <w:rtlGutter w:val="0" />
      …
    </w:sectPr>
  </w:pPr>
</w:p>
…
<w:p>
  <w:pPr>
    <w:sectPr>
      <w:pgMar w:gutter="1440" …/>
      <w:rtlGutter />
      …
    </w:sectPr>
  </w:pPr>
</w:p>
```

The first and second sections both place the gutter on the left side, the first by omission of the rtlGutter attribute, and the second by explicitly turning it off. The third section, however, moves the gutter to the right side via the use of the rtlGutter attribute. *end example*]

If the gutterAtTop element (§17.15.1.50) is specified and true, then each section's gutter is at the top and this setting is ignored.

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.6.17 sectPr (Document Final Section Properties)

This element defines the section properties for the final section of the document. [*Note*: For any other section the properties are stored as a child element of the paragraph element corresponding to the last paragraph in the given section. *end note*]

[*Example*: Consider a document with multiple sections. For all sections except the final section, the sectPr element is stored as a child element of the last paragraph in the section. For the final section, this information is stored as the last child element of the body element, as follows:

```
<w:body>
  <w:p>
  …
  </w:p>
  …
  <w:sectPr>
    (final section's properties)
  </w:sectPr>
</w:body>
```

end example]

Attributes	Description
rsidDel (Section Deletion Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was deleted from the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Section Addition Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was added to the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.50).
rsidRPr (Physical Section Mark Character Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidSect (Section Properties Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SectPr) is located in §A.1. *end note*]

17.6.18 sectPr (Section Properties)

This element defines the section properties for the a section of the document. [*Note*: For the last section in the document, the section properties are stored as a child element of the body element. *end note*]

[*Example*: Consider a document with multiple sections. For all sections except the final section, the sectPr element is stored as a child element of the last paragraph in the section, as follows:

```
<w:body>
  <w:p>
    <w:pPr>
      <w:sectPr>
        (final section's properties)
      </w:sectPr>
    </w:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    </w:p>
    …
    <w:sectPr>
      (final section's properties)
    </w:sectPr>
  </w:body>
```

end example]

Attributes	Description
rsidDel (Section Deletion Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was deleted from the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Section Addition Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was added to the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidRPr (Physical Section Mark Character Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.50).
rsidSect (Section Properties Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SectPr) is located in §A.1. *end note*]

17.6.19 sectPr (Previous Section Properties)

When specified as a child element of sectPrChange, the sectPr element specifies a set of section properties that were modified when the document was set to track all revisions.

[*Example*: If the page orientation was changed with revision tracking enabled, the following WordprocessingML defines the contents of that change:

```
<w:sectPr>
  …
  <w:sectPrChange …>
    <w:sectPr>
      <w:pgSz w:w="15840" w:h="12240"/>
    </w:sectPr>
  </w:sectPrChange>
</w:sectPr>
```

The properties that were changed as part of this revision are stored in this sectPr element. *end example*]

Attributes	Description
rsidDel (Section Deletion Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was deleted from the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidR (Section Addition Revision ID)	Specifies a unique identifier used to track the *editing session* when the section mark for this section was added to the document. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidRPr (Physical Section Mark Character Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
rsidSect (Section Properties Revision ID)	Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted. All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session. A producer can choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SectPrBase) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.6.20 textDirection (Text Flow Direction)

This element specifies the direction of the text flow for this section.

[*Example*: Consider a document with a section in which text be oriented vertically, flowing from left to right horizontally on the page. This setting requires the following WordprocessingML:

```
<w:sectPr>
  …
  <w:textDirection w:val="lr" />
</w:sectPr>
```

The textDirection element specifies via the `lr` value in the val attribute that the text flow be oriented vertically, with subsequent lines stacked from left to right. *end example*]

Attributes	Description

val (Direction of Text Flow)	Specifies the direction of the text flow for this object.
	
	[*Example*: Consider a document with a section in which text must be oriented vertically, flowing from left to right horizontally on the page. This setting requires the following WordprocessingML:
	
	`<w:sectPr>`
	`  …`
	`  <w:textDirection w:val="lr" />`
	`</w:sectPr>`
	
	The textDirection element specifies via the `lr` value in the val attribute that the text flow must be oriented vertically, with subsequent lines stacked from left to right.*end example*]
	
	The possible values for this attribute are defined by the ST_TextDirection simple type (§17.18.93).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextDirection) is located in §A.1. *end note*]

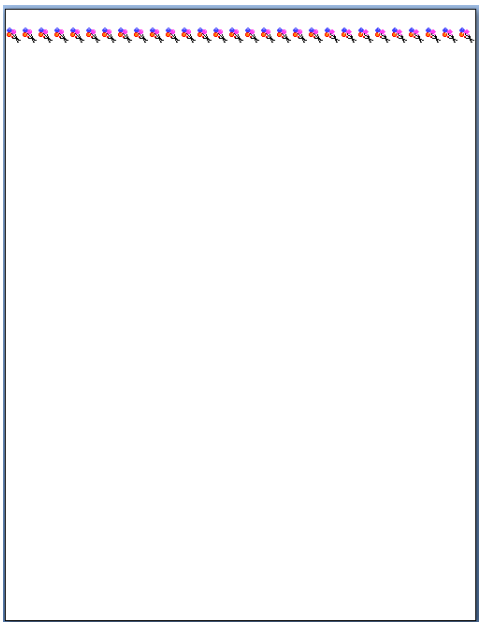
17.6.21 top (Top Border)

This element specifies the presentation and display of the page border displayed at the top of each page in this section.

[*Example*: Consider a section in which all pages must have a top border consisting of a repeated image of balloons, like this:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



This border would result in the following WordprocessingML:

```
<w:sectPr>
  …
  <w:pgBorders>
    <w:top w:val="balloons3Colors" …/>
  </w:pgBorders>
  …
</w:sectPr>
```

Because the page only has a border at the top, only the top element is specified within the set of page borders. *end example*]

When a document has a top border that is relative to the page edges (using an offsetFrom attribute value of page on pgBorders), it shall span the top edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding left or right page border (if one is specified)
- It reaches the edge of the page.

[*Example*: In the example above, no left or right border was specified in the WordprocessingML, so a consumer must draw the border from one edge of the page to the other. *end example*]

When a document has a top border that is relative to the text (using the offsetFrom attribute value of `text` on pgBorders), it shall only span the necessary width to satisfy the requirement of spanning the width of the text.

When a document has custom border art specified by attributes topLeft, topRight, and/or id, it shall use the corresponding relationship part item as an image for the top left corner, top right corner, and/or top border,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

respectively. If the corresponding relationship part item cannot be resolved the consumer shall use the border specified by the value of the val attribute. If the corresponding value of the val attribute cannot be resolved no top left corner, top right corner, or top border is present when the page is displayed.

When a document has a custom border art specified by attribute id without specifying either attributes topRight and/or topLeft, the top border as resolved by the corresponding relationship part item of attribute id shall span to the corners not specified by topRight and/or topLeft attributes.

Attributes	Description
color (Border Color)	Specifies the color for this border. This value can be defined as either: • A color value using the RGB color model whose red, green, and blue values are written as numbers in the range 0 to 255, hex encoded, and concatenated. [*Example*: Full intensity red would be 255 red, 0 green, 0 blue, encoded to FF, 00, 00, and concatenated to `FF0000`. *end example*]. RGB colors are specified in the sRGB color space. • `auto` to allow a consumer to automatically determine the border color in order to make the document's text readable. [*Example*: A document with white text and a background color of `auto` might result in the use of a black background, in order to ensure legibility of the content. *end example*] [*Example*: Consider a border color with value `auto`, as follows: `<w:bottom … w:color="auto"/>` This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*] If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value. The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).
frame (Create Frame Effect)	Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text. If this attribute is omitted, then the border is not given any frame effect. [*Example*: Consider a bottom border which must appear with a frame effect, which is specified in the following WordprocessingML: `<w:bottom w:frame="true" … />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	This frame's val is `true`, indicating that the border frame effect must be applied. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
id (Custom Defined Border Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which contains the custom border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom bottom border in a document: `<w:bottom w:val="custom" r:id="rIdCustomBottomBorder" …/>` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomBottomBorder` must contain the custom bottom border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
shadow (Border Shadow)	Specifies whether this border should be modified to create the appearance of a shadow. For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location. If this attribute is omitted, then the border is not given the shadow effect. [*Example*: Consider a top border which must appear with a shadow effect, resulting in the following WordprocessingML: `<w:bottom w:shadow="true" … />` This frame's val is `true`, indicating that the shadow effect must be applied to the border. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
space (Border Spacing	Specifies the spacing offset that shall be used to place this border on the parent object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

Measurement)	When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§17.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points.
	
	When a document has a page border that is relative to the text extents (using a value of text in the offsetFrom attribute on pgBorders (§17.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.
	
	[*Example*: Consider a document with a set of page borders all specified to appear 24 points from the edge of the page. The resulting WordprocessingML would be as follows:
	
	`<w:pgBorders w:offsetFrom="page">`
	`  <w:bottom … w:space="24" />`
	`</w:pgBorders`
	
	The offsetFrom attribute specifies that the space value provides the offset of the page border from the page edge, and the value of the space attribute specifies that the page offset must be 24 points. *end example*]
	
	The possible values for this attribute are defined by the ST_PointMeasure simple type (§17.18.68).

sz (Border Width)	Specifies the width of the current border.
	
	If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range can be reassigned to a more appropriate value.
	
	If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range can be reassigned to a more appropriate value.
	
	[*Example*: Consider a document with a three point wide dashed line border on all sides, resulting in the following WordprocessingML markup:
	
	`<w:top w:val="dashed" w:sz="24" …/>`
	`<w:left w:val="dashed" w:sz="24" …/>`
	`<w:bottom w:val="dashed" w:sz="24" …/>`
	`<w:right w:val="dashed" w:sz="24" …/>`
	
	The border style is specified using the val attribute, and because that border style is a line border (dashed), the sz attribute specifies the size in eighths of a point (24 eighths of a point = 3 points). *end example*]
	
	The possible values for this attribute are defined by the ST_EighthPointMeasure simple

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	type (§17.18.23).

themeColor (Border Theme Color)	Specifies the base theme color used to generate the border color. The border color is the RGB value associated with themeColor as further transformed by themeTint or themeShade (if one is present), else the background color is the RGB value associated with themeColor.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part (§14.2.7 and §20.1.6.9), which allows color information to be set centrally in the document.
	
	To determine the color to display, the following actions are performed:
	• Using the mapping specified in the ST_ThemeColor simple type (§17.18.97), the appropriate attribute on the clrSchemeMapping element (§17.15.1.20) is read.
	• Using that value and the mapping specified in the ST_ColorSchemeIndex simple type (§17.18.103), the appropriate element in the document's Theme part is read to get the base theme color.
	• The specified color is modified based on the presence of the themeTint or themeShade attribute.
	
	[*Example*: Consider a set of borders configured to use the accent2 theme color, resulting in the following WordprocessingML markup:
	
	`<w:top … w:themeColor="accent2" w:themeTint="99" />`
	`<w:bottom … w:themeColor="accent2" w:themeTint="99" />`
	`<w:left … w:themeColor="accent2" w:themeTint="99" />`
	`<w:right … w:themeColor="accent2" w:themeTint="99" />`
	
	If the Settings part contained the following markup:
	
	`<w:clrSchemeMapping … w:accent2="accent2"/>`
	
	and the Theme part contained the following XML markup:
	
	`<a:accent2>`
	`  <a:srgbClr val="4F81BD"/>`
	`</a:accent2>`
	
	the resulting border color would be 95B3D7 (the result of a 60% tint applied to the original theme color; see the calculations in themeTint below for details). *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Border Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, the value of this attribute shall be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border. [*Example*: Consider a shade of 40% applied to a border in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 \\ = 102 \\ = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$\mathrm{L}' = \mathrm{L} * \mathrm{Shade}_{\mathrm{percentage}}$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the `accent2` theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The equivalent HSL color value would be $\left(\frac{1}{360}, 0.48, 0.53\right)$. Applying the shade formula with a shade percentage of 75% to the luminance, we get: $$L' = 0.53 * 0.75 \\ = 0.39698$$ Taking the resulting HSL color value of $\left(\frac{1}{360}, 0.48, 0.39698\right)$ and converting back to RGB, we get `943634`. This transformed value can be seen in the resulting background's color attribute: `<w:top w:val="single" w:sz="4" w:space="24" w:color="943634" w:themeColor="accent2" w:themeShade="BF"/>` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Border Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this border instance. If the themeColor attribute is not present, then this attribute shall not be used. If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border. The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border. [*Example*: Consider a tint of 60% applied to a border in a document. This tint is calculated as follows: $$T_{xml} = 0.6 * 255 = 153 = 99(hex)$$ The resulting themeTint value in the file format would be 99. *end example*] Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows: • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $$L' = L * \text{Tint}_{pct} + (1 - \text{Tint}_{pct})$$ • Convert the resultant HSL color to RGB [*Example*: Consider a document with a background using the accent2 theme color, whose RGB value (in RRGGBB hex format) is 4F81BD. The equivalent HSL color value would be $\left(\frac{213}{360}, 0.45, 0.53\right)$. Applying the tint formula with a tint percentage of 60% to the luminance, we get: $$L' = 0.53 * 0.6 + (1 - .6) = 0.71$$ Taking the resulting HSL color value of $\left(\frac{213}{360}, 0.45, 0.71\right)$ and converting back to RGB, we get 95B3D7. This transformed value can be seen in the resulting background's color attribute:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:top w:val="single" w:sz="4" w:space="24" w:color="95B3D7" w:themeColor="accent2" w:themeTint="99"/> *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
topLeft (Custom Defined Top Left Border Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship which contains the custom top left border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom top left border shall be used. [*Example*: Consider the following WordprocessingML markup for a custom top left border in a document: <w:top w:val="custom" r:topLeft="rIdCustomTopLeftBorder" …/> The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomTopLeftBorder` must contain the custom top left border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
topRight (Custom Defined Top Right Border Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship which contains the custom top right border image for the parent element. This custom border image is contained in a separate part within the WordprocessingML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. If this attribute is omitted, then no custom top left border shall be used when the parent element is instantiated. [*Example*: Consider the following WordprocessingML markup for a custom top right border in a document: <w:top w:val="custom" r:topRight="rIdCustomTopRightBorder" … />

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rIdCustomTopRightBorder` must contain the custom top right border image for the document. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
val (Border Style)	Specifies the style of border used on this object. This border can either be an art border (a repeated image along the borders - shall only be used for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style. [*Example*: Consider a left border resulting in the following WordprocessingML: `<w:left w:val="single" …/>` This border's val is `single`, indicating that the border style is a single line. *end example*] The possible values for this attribute are defined by the ST_Border simple type (§17.18.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TopPageBorder) is located in §A.1. *end note*]

17.6.22 type (Section Type)

This element specifies the section type of the current section. The section type specifies how the contents of the current section shall be placed relative to the previous section.

WordprocessingML supports five distinct types of section breaks:

- *Next page section breaks* (the default if type is not specified), which begin the new section on the following page.
- *Odd page section breaks*, which begin the new section on the next odd-numbered page.
- *Even page section breaks*, which begin the new section on the next even-numbered page.
- *Continuous section breaks*, which begin the new section on the following paragraph. This means that continuous section breaks might not specify certain page-level section properties, since they shall be inherited from the following section. These breaks, however, can specify other section properties, such as line numbering and footnote/endnote settings.
- *Column section breaks*, which begin the new section on the next column on the page.

Attributes	Description
val (Section Type	Specifies the section type of the current section.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

Setting)	[*Example*: Consider a section that must start on the next page in the document. The WordprocessingML specifying this would look like:
	
	`<w:sectPr>`
	`…`
	`<w:type w:val="nextPage"/>`
	`</w:sectPr>`
	
	The nextPage value specifies that this section starts on the next page. *end example*]
	
	The possible values for this attribute are defined by the ST_SectionMark simple type (§17.18.77).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SectType) is located in §A.1. *end note*]

17.6.23 vAlign (Vertical Text Alignment on Page)

This element specifies the vertical alignment for text on pages in the current section, relative to the top and bottom margins in the main document story on each page.

[*Example*: Consider a section used as a title page on which text must be vertically centered. In order to center the text vertically on the page, the following WordprocessingML is used:

```
<w:sectPr>
  …
  <w:vAlign w:val="center" />
</w:sectPr>
```

The vAlign value of center specifies that text must be laid out in the center of the top and bottom text margins for all pages in this section. *end example*]

Attributes	Description
val (Vertical Alignment Setting)	Specifies the vertical alignment for text between the top and bottom margins of the parent container (page or table cell). [*Example*: Consider a region where the text must be vertically centered in the parent element. This would require a val value of center, in order to specify that all justification vertically must be centered relative to the parent. For a section, this setting would be specified as follows: `<w:vAlign w:val="center" />` The val attribute of center specifies that the content is centered relative to its container

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(in this case, the page). *end example*] The possible values for this attribute are defined by the ST_VerticalJc simple type (§17.18.101).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VerticalJc) is located in §A.1. *end note*]

17.7 Styles

Within a WordprocessingML file, *styles* are predefined sets of table, numbering, paragraph, and/or character properties which can be applied to text within the document. This allows the formatting properties to be stored and managed independently from the content, allowing the look of document content to be changed in a single location (e.g. the look of all first-level headings is changed by changing the style with styleId `Heading1` rather than looking for and changing each paragraph in the document).

[*Example*: The Normal paragraph style in a word processing document can have any number of formatting properties, e.g. font face = Times New Roman; font size = 12pt; paragraph justification = left). All paragraphs which reference this paragraph style would automatically inherit these properties. *end example*]

Each style defined within a WordprocessingML document requires a *style definition*. The style definition contains all of the information needed by a consumer to store and display that style within a WordprocessingML document, and is defined using the style element. The style definition for any style in WordprocessingML can be divided into three segments The complete definition of style properties can be found on the reference for the style element (§17.7.4.17):

- General style properties
- Style types
- Type specific formatting properties

Each of these three segments are discussed in the following subclauses.

17.7.1 Style Inheritance

In order to compile the complete set of paragraph and character properties specified by any given style (as appropriate), a consumer shall follow the rule of style inheritance to determine each property in that set.

Style inheritance states that styles of any given style type can inherit from other styles of that style type, and therefore a consumer shall 'build up' the style information by following the inheritance tree. This inheritance is defined via the basedOn element, which specifies the styleId of the parent style.

[*Example*: The "Tristan Test" paragraph style can inherit properties from the "Heading 1" paragraph style, which itself can inherit properties from the "Normal" paragraph style. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

To build up the resulting style, a consumer shall trace the hierarchy (following each basedOn value) back to a style which has no basedOn element (is not based on another style). The resulting style is then constructed by following each level in the tree, applying the specified paragraph and/or character properties as appropriate. When properties conflict, they are overridden by each subsequent level (this includes turning OFF a property set at an earlier level). Properties which are not specified simply do not change those specified at earlier levels.

[*Example*: Consider a character style Green which specifies only that the text color is green, but inherits from another character style Base which defines a font face of Arial, as well as bold:

```
<w:style w:type="character" w:styleId="Green">
  <w:name w:val="Green" />
  <w:basedOn w:val="Base" />
  <w:rPr>
    <w:color w:val="22B14C" />
  </w:rPr>
</w:style>
…
  <w:style w:type="character" w:styleId="Base">
  <w:name w:val="Base" />
  <w:rPr>
    <w:rFonts w:ascii="Arial" w:hAnsi="Arial" />
    <w:b />
  </w:rPr>
</w:style>
```

The definition of the Green character style has a basedOn element which specifies the Base style. This means that any use of the Green style is defined as bold, green, Arial text. *end example*]

Conversely, a producer should not output any property on a style which has already been set by a previous level of the style hierarchy, as well as those which match the document defaults. This means that if the document defaults or any previous level in a style's hierarchy specify a property which is unchanged at this level, that property should not be part of the style definition in the resulting WordprocessingML.

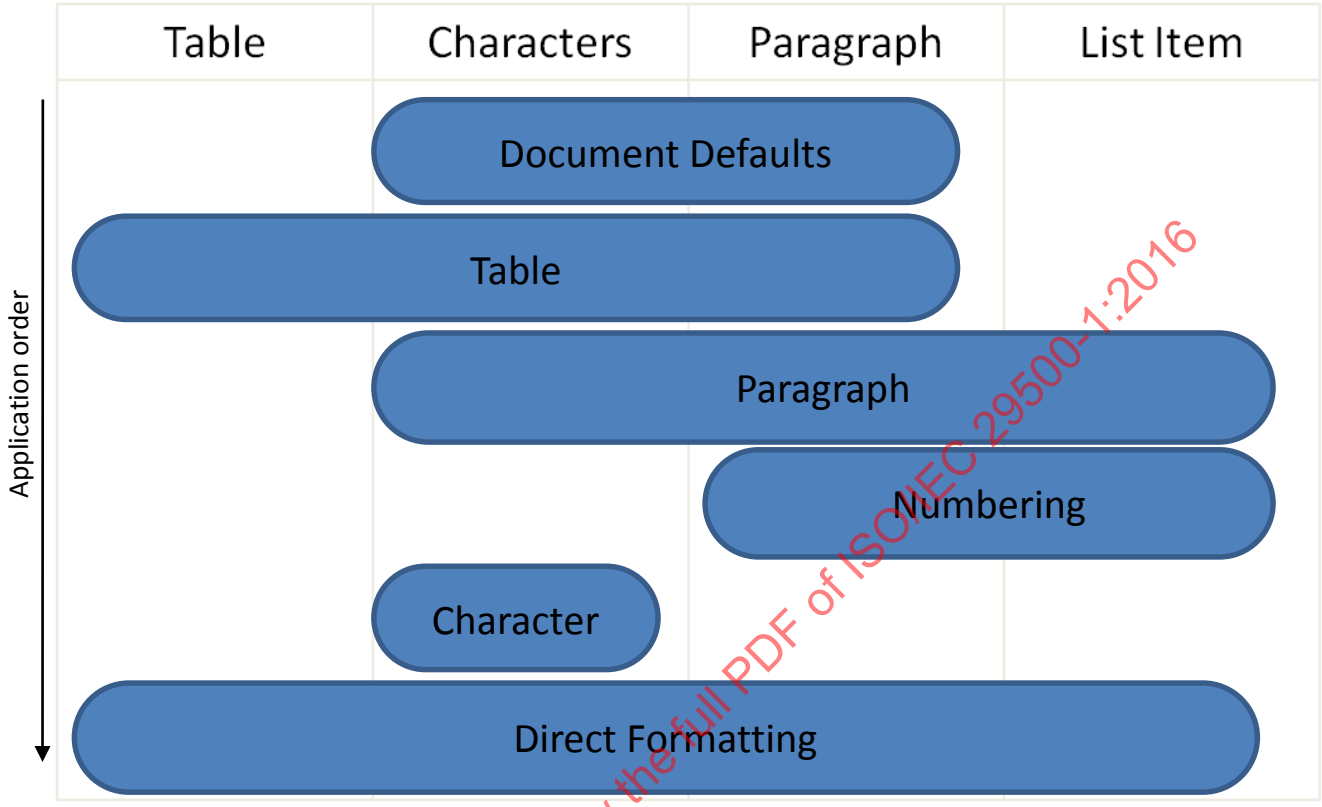
[*Example*: If the document default font is Bauhaus 93 and the Heading 1 style also specifies the Bauhaus 93 font, then a producer should not output any rFonts element for the Heading 1 style definition, because that formatting is inherited from the document defaults. *end example*]

17.7.2 Style Hierarchy

With the various flavors of styles available (see each of the subclauses below), multiple style types can be applied to the same content within a file, which means that properties shall be applied in a specific deterministic order. As with inheritance, the resulting formatting properties set by one style type can be unchanged, removed, or altered by following style types.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The following table illustrates the order of application of these defaults, and which properties are impacted by each:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This process can be described as follows:

- First, the document defaults are applied to all runs and paragraphs in the document.
- Next, the table style properties are applied to each table in the document, following the conditional formatting inclusions and exclusions specified per table.
- Next, numbered item and paragraph properties are applied to each paragraph formatted with a numbering style.
- Next, paragraph and run properties are applied to each paragraph as defined by the paragraph style.
- Next, run properties are applied to each run with a specific character style applied.
- Finally, we apply direct formatting (paragraph or run properties not from styles). If this direct formatting includes numbering, that numbering + the associated paragraph properties are applied.

If the value of the rFonts element (§17.3.2.26) references a font which is not available, applications determine a suitable alternative font via a process called font substitution, which is defined in §17.8.2.

17.7.3 Toggle Properties

Certain character properties defined in §17.3.2 are specified as toggle properties. [*Example*: the Bold and Italics properties are toggle properties. *end example*] As indicated in the previous two sections (§17.7.1 and §17.7.2) several styles can affect the formatting applied to a given piece of content within a WordprocessingML

©ISO/IEC 2016 – All rights reserved

document. When the same formatting property appears in one or more styles that affect the content applied to a run, the combined effect depends on whether or not the formatting property is a toggle property.

If the property is not a toggle property, then its values shall be applied in the order described in §17.7.1 and §17.7.2, and only its last value in that order shall be used.

If the property is a toggle property, then its values, which are limited to `true` and `false` (or the equivalent values `1` and `0`) shall be combined as follows:

- If a toggle property is explicitly set in direct formatting applied to a given piece of content, then its value in the direct formatting shall be used.
- Otherwise, the instances of that toggle property in the styles that affect the content shall be combined in the following manner:
    - If multiple instances of the toggle property appear at the same level of the style hierarchy, then the first value encountered by the following algorithm shall be used (if no value is encountered, the property takes on its default value).
        - Attempt to read the value in the style.
        - If it does not exist and the style has a basedOn element with a non-empty value, repeat step 1 using the style specified by the basedOn element.
        - [*Example*: If a paragraph style sets no value for the bold property to false and the paragraph style specified by its basedOn element specifies that it is true, the result of applying the style definition sets the value of bold to true (the first value in the hierarchy). *end example*]
    - If the value of the toggle property appears at multiple levels of the style hierarchy (§17.7.2), their effective values shall be combined as follows:
        - If the value specified by the document defaults is true, the effective value is true.
        - Otherwise, the values are combined by a Boolean XOR as follows:
          $value_{effective} = val_{table}\ XOR\ val_{paragraph}\ XOR\ val_{character}$
          i.e., the effective value to be applied to the content shall be true if its effective value is true for an odd number of levels of the style hierarchy.

The following Boolean properties are toggle properties: §17.3.2.1 (Bold), §17.3.2.2 (Complex Script Bold), §17.3.2.5 (Display All Characters as Capital Letters), §17.3.2.13 (Embossing), §17.3.2.16 (Italics), §17.3.2.17 (Complex Script Italics), §17.3.2.18 (Imprinting), §17.3.2.23 (Display Character Outline), §17.3.2.31 (Shadow), §17.3.2.33 (Small Caps), §17.3.2.37 (Single Strikethrough), §17.3.2.41 (Hidden Text).

[*Example*: Consider a table style with two styles in its basedOn chain. If the resolved value of the bold property (a toggle property) within the basedOn chain of the table style is true, that specifies that this property should be applied to the contents of the table:

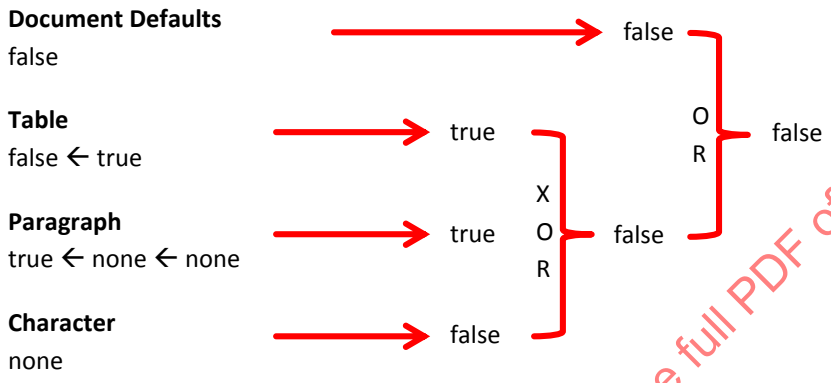
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Cell 1,1	**Cell 1,2**
Cell 2,1	**Cell 2,2**

If a single paragraph within that table also has a paragraph style applied, with three styles in its basedOn chain that resolve to a value of true, the toggle property logic above would toggle the bold property, resulting in bold not being applied to its contents. Applying this to the paragraph in the first cell below, the resulting table would appear as follows:

Cell 1,1	**Cell 1,2**
Cell 2,1	**Cell 2,2**

The calculation which results in this value for the bold property is displayed below:



← symbolizes the traversal of a basedOn reference

end example]

17.7.4 General Style Properties

General style properties refer to the set of properties which can be used regardless of the type of style. [*Example*: Within a style definition the style name, additional aliases for the style, a style ID (used by the document content to refer to the style), if style is hidden, if style is locked, etc. are general style properties. *end example*]

[*Example:* Consider a style called Heading 1 in a document as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:style w:type="paragraph" w:styleId="Heading1">
  <w:name w:val="Heading 1"/>
  <w:basedOn w:val="Normal"/>
  <w:next w:val="Normal"/>
  <w:link w:val="Heading1Char"/>
  <w:uiPriority w:val="1"/>
  <w:qFormat/>
  <w:rsid w:val="00F303CE"/>
  …
</w:style>
```

Above the formatting information specific to this style type are a set of general style properties which define information shared by all style types. *end example*]

17.7.4.1 aliases (Alternate Style Names)

This element specifies the set of alternative names for the parent style definition. These names can be used in an application's user interface as desired. The alternate names shall be stored in this element's val attribute, and each name shall be separated by one or more consecutive comma characters (Unicode character value 002C). All commas present shall be interpreted as separator character and never as part of an alternate style name.

If present, the alternate style names shall be used in the user interface in place of the built-in name specified in the name element (§17.7.4.9) when the appropriate value is set in the stylePaneFormatFilter element (§17.15.1.86).

If this element is omitted, then the style shall not have any alternate style names.

[*Example*: Consider a style with a primary name and two alternate names, defined using the name and aliases elements, as follows:

```
<w:style w:styleId="TestStyle" … >
  <w:name w:val="GD20Complex"/>
  <w:aliases w:val="Regional Growth,Complex Growth"/>
  …
</w:style>
```

This style specifies that it has the primary name `GD20Complex` using the name element (§17.7.4.9), as well as two alternate names `Regional Growth` and `Complex Growth` using the aliases element. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.7.4.2 autoRedefine (Automatically Merge User Formatting Into Style Definition)

This element specifies whether an application shall automatically modify this style when the contents of an entire paragraph in the document with this style applied are modified, ensuring that although only a single instance of text with this style was modified, that change is stored on the style and therefore propagated to all locations where the style is in use.

If this element is omitted, then formatting shall not automatically be merged back into the style definition.

[*Example*: Consider a style defined as follows in a WordprocessingML document:

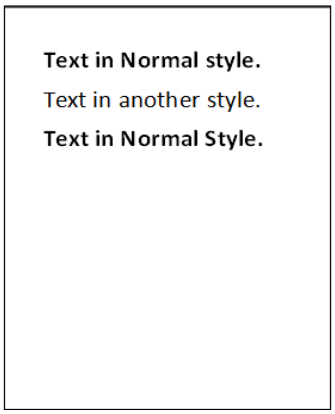
```
<w:style w:styleId="Normal" … >
  <w:name w:val="Normal"/>
  <w:autoRedefine/>
  <w:rPr>
    <w:b/>
  </w:rPr>
  …
</w:style>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

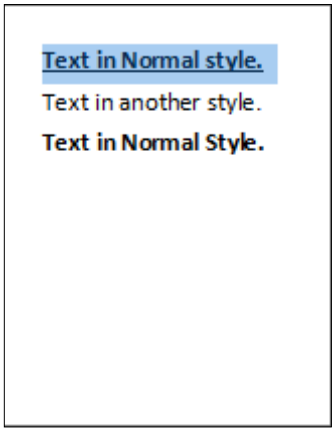
 ©ISO/IEC 2016 – All rights reserved

This style specifies via the use of the autoRedefine element that any formatting applied to text which uses this style must be merged back into the style definition (assuming, of course, that this is a paragraph style).

For example, consider a document which uses the Normal style as defined above:



The first and third paragraphs use the Normal style, and hence have the bold property applied. If an application were to add the underline formatting to the entire first paragraph, as follows:



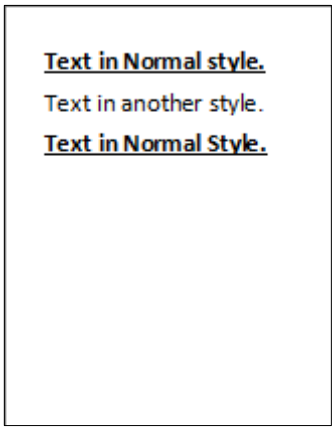
That property, rather than being saved as direct formatting, must be used to update the associated Normal style to add this property, specified using the u element (§17.3.2.40).

```
<w:style w:styleId="Normal" … >
  <w:name w:val="Normal"/>
  <w:autoRedefine/>
  <w:rPr>
    <w:b/>
    <w:u/>
  </w:rPr>
  …
</w:style>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Since this property is automatically merged into the style, it would also appear on the third paragraph (note that the step above would normally be automatically modified into the state shown below, and not discrete as shown above).



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.3 basedOn (Parent Style ID)

This element specifies the style ID of the parent style from which this style inherits in the style inheritance. The *style inheritance* refers to a set of styles which inherit from one another to produce the resulting set of properties for a single style. The val attribute of this element specifies the styleId attribute for the parent style in the style inheritance.

If this element is omitted, then this style shall not be based on any other style in the current document (i.e. this element is the root of the style inheritance for a style). If no style in the current document specifies the styleId present in the val attribute, then this element shall be ignored (i.e. this element is the root of the style inheritance for a style).

If a style with this styleId is present, then it shall be subject to the following restrictions:

- If the current style is a table style, then the parent style shall also be a table style, or this element shall be ignored.
- If the current style is a paragraph style, then the parent style shall also be a paragraph style, or this element shall be ignored.
- If the current style is a character style, then the parent style shall also be a character style, or this element shall be ignored.
- If the current style is a numbering style, then this element shall be ignored.

[*Example*: Consider three WordprocessingML character styles defined as follows:

- A character style with a styleId value of `Strong` whose properties consist of the bold property
- A character style with a styleId value of `Underline` whose properties consist of the underline property

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- A character style with a styleId value of `Emphasis` whose properties consist of the italics property

Each of these character styles defines a single character formatting property. If the basedOn values for each element were defined as follows:

```
<w:style w:styleId="Strong">
  <w:basedOn w:val="Underline"/>
  …
  <w:rPr>
    <w:b/>
  </w:rPr>
</w:style>
<w:style w:styleId="Underline">
  <w:basedOn w:val="Emphasis"/>
  …
  <w:rPr>
    <w:u/>
  </w:rPr>
</w:style>
<w:style w:styleId="Emphasis">
  …
  <w:rPr>
    <w:i/>
  </w:rPr>
</w:style>
```

The `Strong` style is based on the `Underline` style which is in turn based on the `Emphasis` style. This means that the actual definition of the `Strong` style would be as follows:

- Bold
- Underline (inherited from `Underline`)
- Italics (inherited from `Emphasis`)

The style chain for the `Strong` style would be defined as follows:

- `Emphasis`
- `Underline`
- `Strong`

Similarly, the style chain for the `Underline` style would be defined as follows:

- `Emphasis`
- `Underline`

In each case, the style chain is the list of all styles which are combined in order to produce the entire set of properties for any given style. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.7.4.4 hidden (Hide Style From User Interface)

This element specifies whether this style shall be hidden from any and all user interfaces when this document is loaded by an application. If this element is set, then this style can be used to format content (i.e. any content which references this style shall have its properties as normal), but the style shall be hidden from all user interface associated with that application. [*Note*: This setting is typically used to hide styles which are being used internally by an application which should not be used as formatting in a typical case. *end note*]

If this element is omitted, then the style shall not be required to be hidden from the user interface.

[*Example*: Consider a style with a primary name of `InternalStyle` that should not be displayed in any user interface. This requirement would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:style … w:styleId="Style2">
  <w:name w:val="InternalStyle"/>
  <w:hidden/>
  …
</w:style>
```

The hidden element specifies that this style definition shall be round-tripped with the file (since it is part of the document) but should not be displayed in any user interface associated with an application which processes this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.5 latentStyles (Latent Style Information)

This element specifies the properties which shall be applied to a set of latent styles for this document. *Latent styles* refer to any set of style definitions known to an application which have not been included in the current document. [*Example*: Latent styles can include additional styles known by a particular hosting application. *end example*]

When a style definition is embedded in a document, it specifies two distinct groups of properties:

- Behavior properties
- Formatting properties

Obviously, embedding all the styles known to a particular application in each document which it produces would drastically increase the file size. Latent styles provide a way to store pieces of information for the first group (behavior properties) which shall be specified for all styles known to an application without requiring the storage of the second group (formatting properties).

If this element is omitted, the values of the settings represented by each of its attributes are given by the defaults in the attribute descriptions below.

[*Example*: Consider a WordprocessingML document which contains text specified in one of two styles: `Heading1` or `Normal`. Based on this, the document only needs to store the formatting properties for those two styles, saving the additional overhead which would be required to save all of the styles supported by the hosting application.

However, if the documentProtection element (§17.15.1.29) specifies that the hosting application shall prevent the use of any style whose locked element (§17.7.4.7) is set to `false`, then the locking state of all styles known to that application become useful and necessary to maintain the current state of the document. Using latent styles, this information can be stored without storing any formatting properties for those styles.

For example, if all styles which are not stored in the document must be locked except for the style with a primary name (§17.7.4.9) of `Heading 2`. This requirement would be specified using latent styles as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:latentStyles … w:defLockedState="true">
  <w:lsdException w:name="Heading 2" w:locked="false"/>
</w:latentStyles>
```

The latentStyles element specifies that all latent styles known to any hosting application must have a default locking state of `true` except for any style known to the hosting application with a primary name of `Heading 2`, whose latent style definition specifies that its locked state must be `false`. *end example*]

Attributes	Description

count (Latent Style Count)	Specifies the number of known styles which shall be initialized to the current latent style defaults when this document is first processed. [*Note*: This property can be used by an application as needed to ensure that only the number of styles known when this document was created are initialized with the defaults on the parent element, and that all new known styles use their default values. *end note*]
	
	[*Example*: Consider a WordprocessingML document in which only the first `20` latent styles must be initialized. This requirement would be specified as follows:
	
	`<w:latentStyles w:count="20" … >`
	`…`
	`</w:latentStyles>`
	
	The count attribute specifies that 20 known styles must be initialized to the default settings when the document is first opened, and any additional styles should use the defaults defined by the application. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

defLockedState (Default Style Locking Setting)	Specifies the default setting for the locked element (§17.7.4.7) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§17.7.4.8) exists.
	
	If this attribute is omitted, the default locked state for all latent styles in the current document shall be `false`.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must be locked. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defLockedState="true">`
	`…`
	`</w:latentStyles>`
	
	The defLockedState attribute specifies that all latent styles in the current document must have a locked element setting of `true` by default. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

defQFormat (Default Primary Style Setting)	Specifies the default setting for the qFormat element (§17.7.4.14) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§17.7.4.8) exists.
	
	If this attribute is omitted, the default qFormat state for all latent styles in the current document shall be `false`.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must not be marked as primary styles. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defQFormat="false">`
	`…`
	`</w:latentStyles>`
	
	The defQFormat attribute specifies that all latent styles in the current document must have a qFormat element setting of `false` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

defSemiHidden (Default Semi-Hidden Setting)	Specifies the default setting for the semiHidden element (§17.7.4.16) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§17.7.4.8) exists.
	
	If this attribute is omitted, the default semiHidden state for all latent styles in the current document shall be `false`.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must not be marked as semi-hidden. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defSemiHidden="false">`
	`…`
	`</w:latentStyles>`
	
	The defSemiHidden attribute specifies that all latent styles in the current document must have a semiHidden element setting of `false` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

defUIPriority (Default User Interface Priority Setting)	Specifies the default setting for the uiPriority element (§17.7.4.19) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§17.7.4.8) exists.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute is omitted, the default uiPriority state for all latent styles in the current document shall be 99.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must not be marked as semi-hidden. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defUIPriority="10">`
	`  …`
	`</w:latentStyles>`
	
	The defUIPriority attribute specifies that all latent styles in the current document must have a uiPriority element setting of `10` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

defUnhideWhenUsed (Default Hidden Until Used Setting)	Specifies the default setting for the unhideWhenUsed element (§17.7.4.20) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§17.7.4.8) exists.
	
	If this attribute is omitted, the default unhideWhenUsed state for all latent styles in the current document shall be `false`.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must be hidden until they are used in the document's contents. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defUnhideWhenUsed="true">`
	`  …`
	`</w:latentStyles>`
	
	The defUnhideWhenUsed attribute specifies that all latent styles in the current document must have a unhideWhenUsed element setting of `true` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LatentStyles) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.7.4.6 link (Linked Style Reference)

This element specifies the pairing of styles which comprise a linked style. A *linked style* is a grouping of a paragraph style and character style which is used in a user interface to allow the same set of formatting properties to be applied:

- To the contents of one or more entire paragraphs (i.e. as a paragraph style)
- To the contents of one or more runs within a paragraph (i.e. as a character style)

Each style continues to exist independently in the file format as there is both a paragraph and character style present within the styles element (§17.7.4.18), however these two styles shall be merged into one and applied appropriately based on whether they are applied to run(s) or paragraph(s), by referencing the styleId attribute of the paired linked style via this element's val attribute.

A style element without a child link element is not part of a linked style pairing. If no style in the current document specifies the styleId present in the val attribute, then this element shall be ignored.

If a style with this styleId is present, then it shall be subject to the following restrictions:

- If the parent style is a table style, then this element shall be ignored.
- If the parent style is a paragraph style, then this element's val attribute must refer to a character style, or this element shall be ignored.
- If the parent style is a character style, then this element's val attribute must refer to a paragraph style, or this element shall be ignored.
- If the parent style is a numbering style, then this element shall be ignored.

[*Example*: Consider a linked style defined as follows in a WordprocessingML document:

```
<w:style w:type="paragraph" w:styleId="TestParagraphStyle">
  <w:link w:val="TestCharacterStyle"/>
  …
</w:style>
<w:style w:type="character" w:styleId="TestCharacterStyle">
  <w:link w:val="TestParagraphStyle"/>
  …
</w:style>
```

This pairing of a paragraph style and a character style are linked via the link element, which is used to reference the styleId of the paragraph style from the character style definition and vice versa. Because this pairing is permitted based on the rules above, the resulting combination must be used as a linked style, which appears as one style in an application, but uses the character and/or paragraph style as appropriate. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.7.4.7 locked (Style Cannot Be Applied)

This element specifies whether an application shall prevent the use of this style when this document is loaded and/or modified. If this element is set, then this style can be used to format existing content (i.e. any content which references this style shall have its properties as normal), but new instances of the style shall be prevented from being applied via all mechanisms associated with that application.

If this element is omitted, then the use of the style shall not be prevented by an application processing this document.

[*Example*: Consider a style with a primary name of `Test Style` which should be locked, and prevented from being added to any content in a given document. This requirement would be specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:style … w:styleId="TestStyle">
  <w:name w:val="Test Style"/>
  <w:locked/>
  …
</w:style>
```

The presence of the locked element specifies that new instances of the style must be prevented from being applied via all mechanisms associated with that application. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.8 lsdException (Latent Style Exception)

This element specifies the properties which shall be applied a single latent style for this document. *Latent styles* refer to any set of known style definitions which have not been included in the current document.

[*Example*: Consider a WordprocessingML document which contains text specified in one of two styles: `Heading1` or `Normal`. Based on this, the document only needs to store the formatting properties for those two styles, saving the additional overhead which would be required to save all of the styles supported by the hosting application.

However, if the documentProtection element (§17.15.1.29) specifies that the hosting application must prevent the use of any style whose locked element (§17.7.4.7) is set to `false`, then the locking state of all styles known to that application become useful and necessary to maintain the current state of the document. Using latent styles, this information can be stored without storing any formatting properties for those styles.

For example, if all styles which are not stored in the document must be locked except for the style with a primary name (§17.7.4.9) of `Heading 2`. This requirement would be specified using latent styles as follows:

```
<w:latentStyles … w:defLockedState="true">
  <w:lsdException w:name="Heading 2" w:locked="false"/>
</w:latentStyles>
```

The lsdException element specifies that the latent style with a primary name of `Heading 2` must have a locked state setting of `false`. *end example*]

If this element is omitted, there are no latent style exceptions to the defaults specified on the latentStyles element.

Attributes	Description
locked (Latent Style Locking Setting)	Specifies the default setting for the locked element (§17.7.4.7) which shall be applied to the latent style with the matching style name value. If this attribute is omitted, the default locked state for this latent style shall be determined by the defLockedState attribute on the parent latentStyles element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must be locked except for the `TestStyle` style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defLockedState="true">`
	`  <w:lsdException w:name="TestStyle" w:locked="false"/>`
	`</w:latentStyles>`
	
	The locked attribute on the latent style exception specifies that the `TestStyle` style must have a locked element setting of `false` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

name (Primary Style Name)	Specifies the primary name for the style which shall inherit this set of latent style property exceptions.
	
	If the current application does not know of an internal primary style with the current name, then this set of latent style exceptions can be ignored.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must be locked except for the `TestStyle` style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defLockedState="true">`
	`  <w:lsdException w:name="TestStyle" w:locked="false"/>`
	`</w:latentStyles>`
	
	The name attribute on the latent style exception specifies that the `TestStyle` style must have this set of latent style properties (if the application knows of a style with this name). *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

qFormat (Latent Style Primary Style Setting)	Specifies the default setting for the qFormat element (§17.7.4.14) which shall be applied to the latent style with the matching style name value.
	
	If this attribute is omitted, the default qFormat state for this latent style shall be determined by the defQFormat attribute on the parent latentStyles element.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must not be primary styles except for the `TestStyle` style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defQFormat="false">`
	`  <w:lsdException w:name="TestStyle" w:qFormat="true"/>`
	`</w:latentStyles>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The qFormat attribute on the latent style exception specifies that the TestStyle style must have a qFormat element setting of true by default. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

semiHidden (Semi hidden text override)	Specifies the default setting for the semiHidden element (§17.7.4.16) which shall be applied to the latent style with the matching style name value.
	
	If this attribute is omitted, the default semiHidden state for this latent style shall be determined by the defSemiHidden attribute on the parent latentStyles element.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must not be semi-hidden except for the TestStyle style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defSemiHidden="false">`
	`  <w:lsdException w:name="TestStyle" w:semiHidden="true"/>`
	`</w:latentStyles>`
	
	The semiHidden attribute on the latent style exception specifies that the TestStyle style must have a semiHidden element setting of true by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

uiPriority (Override default sorting order)	Specifies the default setting for the uiPriority element (§17.7.4.19) which shall be applied to the latent style with the matching style name value.
	
	If this attribute is omitted, the default uiPriority state for this latent style shall be determined by the defUIPriority attribute on the parent latentStyles element.
	
	[*Example*: Consider a WordprocessingML document in which all styles which are not stored in the document must have a priority value of 10 except for the TestStyle style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defUIPriority="10">`
	`  <w:lsdException w:name="TestStyle" w:uiPriority="25"/>`
	`</w:latentStyles>`
	
	The uiPriority attribute on the latent style exception specifies that the TestStyle style must have a uiPriority element setting of 25 by default. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

unhideWhenUsed (Unhide when used)	Specifies the default setting for the unhideWhenUsed element (§17.7.4.20) which shall be applied to the latent style with the matching style name value. If this attribute is omitted, the default unhideWhenUsed state for this latent style shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	be determined by the defUnhideWhenUsed attribute on the parent latentStyles element.
	
	[*Example*: Consider a WordprocessingML document in which all styles are to be hidden until used except for the `TestStyle` style. This requirement would be specified using latent styles as follows:
	
	`<w:latentStyles … w:defUnhideWhenUsed="true">`
	`  <w:lsdException w:name="TestStyle" w:unhideWhenUsed="false"/>`
	`</w:latentStyles>`
	
	The unhideWhenUsed attribute on the latent style exception specifies that the `TestStyle` style must have an unhideWhenUsed element setting of `false` by default. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LsdException) is located in §A.1. *end note*]

17.7.4.9 name (Primary Style Name)

This element specifies the primary name for the current style in the document. This name can be used in an application's user interface as desired. The actual primary name for this style is stored in its val attribute.

If present, the alternate style names (§17.7.4.1) shall be used in the user interface in place of the built-in name specified when the appropriate value is set in the stylePaneFormatFilter element (§17.15.1.85).

If this element is omitted, then the style shall not have a primary style name.

[*Example*: Consider a style with a primary name and two alternate names, defined using the name and aliases elements, as follows:

```
<w:style w:styleId="TestStyle" … >
  <w:name w:val="GD20Complex"/>
  <w:aliases w:val="Regional Growth,Complex Growth"/>
  …
</w:style>
```

This style specifies that it has the primary name `GD20Complex` using the name element. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's styleId.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.7.4.10 next (Style For Next Paragraph)

This element specifies the style which shall automatically be applied to a new paragraph created following a paragraph with the parent paragraph style applied. [*Note*: This setting is typically used when the use of the current style is limited to one paragraph at most, and it would typically be undesirable to apply this style to following paragraphs - for example, a title style might specify that its following paragraphs must return to regular text formatting. *end note*]

If this element is specified on a style of any style type other than a paragraph style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a paragraph style, this element shall be ignored.

If this element is omitted, then the following paragraph shall use the same paragraph style as the current paragraph.

[*Example*: Consider a style defined as follows in a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:style w:styleId="TestParagraphStyle" … >
  <w:name w:val="Test Paragraph Style"/>
  <w:next w:val="AnotherParagraphStyle"/>
  <w:rPr>
    <w:b/>
  </w:rPr>
  …
</w:style>
```

This style specifies via the use of the next element that the style for the next paragraph in the document must be the paragraph style whose styleId attribute value is `AnotherParagraphStyle` (if such a paragraph style exists). *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.7.4.11 personal (E-Mail Message Text Style)

This element specifies that the parent style, when in use in the context of an e-mail message, was used by default to format all message text from one or more users. [*Note*: This setting does not provide any additional semantic about the style, but can be used in the context of e-mail to automatically reformat the contents of the e-mail message while ignoring any content to which styles were deliberately applied (since this style was implicitly applied to message text without user interaction). *end note*]

If this element is specified on a style of any style type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message text style in the context of e-mail messages.

[*Example*: Consider a style defined as follows in a WordprocessingML document:

```
<w:style w:styleId="EmailText" w:type="character" >
  <w:name w:val="EmailText"/>
  <w:personal w:val="true" />
  <w:rPr>
    …
  </w:rPr>
</w:style>
```

This style specifies via the use of the personal element that this style is a style used to format message text in the context of e-mail. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.12 personalCompose (E-Mail Message Composition Style)

This element specifies that the parent style, when in use in the context of an e-mail message, can be used by default to format new message text within the e-mail message. [*Note*: This setting does not provide any additional semantic about the style, but can be used in the context of e-mail to automatically format the contents of new test in the e-mail message. *end note*]

If this element is specified on a style of any style type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message composition text style in the context of e-mail messages.

[*Example*: Consider a style defined as follows in a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:style w:styleId="EmailText" w:type="character" >
  <w:name w:val="EmailText"/>
  <w:personalCompose w:val="true" />
  <w:rPr>
    …
  </w:rPr>
</w:style>
```

This style specifies via the use of the personalCompose element that this style is a style used to format new message text in the context of e-mail. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.13 personalReply (E-Mail Message Reply Style)

This element specifies that the parent style, when in use in the context of an e-mail message, can be used by default to format existing message text within the e-mail message when a new reply is generated. [*Note*: This setting does not provide any additional semantic about the style, but can be used in the context of e-mail to automatically format the contents of existing test in the e-mail message. *end note*]

If this element is specified on a style of any style type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message reply text style in the context of e-mail messages.

[*Example*: Consider a style defined as follows in a WordprocessingML document:

```
<w:style w:styleId="EmailText" w:type="character" >
  <w:name w:val="EmailText"/>
  <w:personalReply w:val="true" />
  <w:rPr>
    …
  </w:rPr>
</w:style>
```

This style specifies via the use of the personalReply element that this style is a style used to format existing message text in the context of e-mail. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.14 qFormat (Primary Style)

This element specifies whether this style shall be treated as a primary style when this document is loaded by an application. If this element is set, then this style has been designated as being particularly important for the current document, and this information can be used by an application in any means desired. [*Note*: This setting

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

does not imply any behavior for the style, only that the style is of particular significance for this document. *end note*]

If this element is omitted, then the style shall not be considered a primary style for this document.

[*Example*: Consider a style with a primary name of `PrimaryStyleExample` that should be treated as a primary style for the document. This requirement would be specified using the following WordprocessingML:

```
<w:style … w:styleId="PStyle">
  <w:name w:val="PrimaryStyleExample"/>
  <w:qFormat/>
  …
</w:style>
```

The qFormat element specifies that this style definition must be treated as a primary style for this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.4.15 rsid (Revision Identifier for Style Definition)

This element specifies a unique four digit number which shall be used to determine the editing session in which this style definition was last modified. This value shall follow this following constraint: All document elements which specify the same rsid* values shall correspond to changes made during the same editing session. An *editing session* is defined as the period of editing which takes place between any two subsequent save actions. [*Note*: This setting does not imply any behavior for the style, only that the style was last modified during one particular editing session. This information can be interpreted by an application in any manner desired. *end note*]

If this element is omitted, then no revision identifier shall be associated with the parent style definition.

[*Example*: Consider a style with a primary name of `PrimaryStyleExample` that is defined as follows:

```
<w:style … w:styleId="PStyle">
  <w:name w:val="PrimaryStyleExample"/>
  <w:rsid w:val="3E412D01"/>
  …
</w:style>
```

The rsid element specifies that this style definition was last edited in the editing session corresponding to the value `3E412D01`. *end example*]

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	[*Example*: Consider the following value for an attribute of simple type ST_LongHexNumber: `00BE2C6C`. This value is permitted, as it contains four hexadecimal digits, each an encoding of an octet of the actual decimal number value. It can therefore be interpreted as desired in the context of the parent XML element, *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LongHexNumber) is located in §A.1. *end note*]

17.7.4.16 semiHidden (Hide Style From Main User Interface)

This element specifies whether this style shall be hidden from the main user interface when this document is loaded by an application. If this element is set, then this style can be used to format content (i.e. any content which references this style shall have its properties as normal), but the style shall be hidden from the main user interface associated with that application.

[*Note*: The interpretation of a "main" user interface must not be dictated by ISO/IEC 29500, and can be defined by an application as appropriate.

This setting is intended to define a style property which allows styles to be seen and modified in an advanced user interface, without exposing the style in a less advanced setting, for example, the style which is used to format the contents of a comment should typically not be shown in a simple user interface (as it is uncommon to want to modify it), but would be inappropriate to hide completely using the hidden element (§17.7.4.4), as very advanced users might want to change its appearance. *end note*]

If this element is omitted, then the style shall not be required to be hidden from the main user interface.

[*Example*: Consider a style with a primary name of `Comment Style` that should not be displayed in the main user interface. This requirement would be specified using the following WordprocessingML:

```
<w:style … w:styleId="CStyle">
  <w:name w:val="Comment Style"/>
  <w:semiHidden/>
  …
</w:style>
```

The semiHidden element specifies that this style definition should not be displayed in any main user interface associated with an application which processes this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.7.4.17 style (Style Definition)

This element specifies the definition of a single style within a WordprocessingML document. A *style* is a predefined set of table, numbering, paragraph, and/or character properties which can be applied to regions within a document.

The style definition for any style definition can be divided into three segments:

- General style properties
- Style type
- Style type-specific properties

General style properties refers to the set of properties which can be used regardless of the style type; for example, the style name, additional aliases for the style, a style ID (used by the document content to refer to the style), if style is hidden, if style is locked, etc.

[*Example:* Consider a style called Heading 1 in a document as follows:

```
<w:style w:type="paragraph" w:styleId="Heading1">
  <w:name w:val="Heading 1"/>
  <w:basedOn w:val="Normal"/>
  <w:next w:val="Normal"/>
  <w:link w:val="Heading1Char"/>
  <w:uiPriority w:val="1"/>
  <w:qFormat/>
  <w:rsid w:val="00F303CE"/>
  …
</w:style>
```

Above the formatting information specific to this style type are a set of general style properties which define information shared by all style types. *end example*]

Style types refers to the property on a style which defines the type of style created with this style definition. WordprocessingML supports six types of style definitions by the values for the style definition's type attribute:

- Paragraph styles
- Character styles
- Linked styles (paragraph + character) [*Note*: Accomplished via the link element (§17.7.4.6). *end note*]
- Table styles
- Numbering styles
- Default paragraph + character properties

[*Example:* Consider a style called Heading 1 in a document as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:style w:type="paragraph" w:styleId="Heading1">
  <w:name w:val="Heading 1"/>
  <w:basedOn w:val="Normal"/>
  <w:next w:val="Normal"/>
  <w:link w:val="Heading1Char"/>
  <w:uiPriority w:val="1"/>
  <w:qFormat/>
  <w:rsid w:val="00F303CE"/>
  …
</w:style>
```

The type attribute has a value of paragraph, which indicates that the following style definition is a paragraph style. *end example*]

Style type-specific properties refers to the payload of the style: its formatting information as well as any properties which apply only to that style type.

[*Example*: Consider a table style with primary name `Normal Table` defined as follows:

```
<w:style w:type="table" w:default="1" w:styleId="TableNormal">
  <w:name w:val="Normal Table"/>
  …
  <w:tblPr>
    <w:tblInd w:w="0" w:type="dxa"/>
    <w:tblCellMar>
      <w:top w:w="0" w:type="dxa"/>
      <w:start w:w="108" w:type="dxa"/>
      <w:bottom w:w="0" w:type="dxa"/>
      <w:end w:w="108" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
</w:style>
```

The tblPr element contains the formatting payload for this table style, which is only applicable to a table style. *end example*]

Attributes	Description
customStyle (User-Defined Style)	Specifies that this style is a user-defined style (i.e. it is not a style which was automatically generated by an application). This setting (specifically a value of true or its equivalents) shall not allow the formatting associated with the style to be changed automatically by an application, but can be used to specify that if the associated style ID is known, certain user interface behaviors can be applied to its definition. [*Example*: The style's primary name can be localized to match the current user interface language. *end example*] If this attribute is omitted, then the style shall be assumed to be a built-in style.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider a paragraph style defined as follows: `<w:style w:type="paragraph" w:styleId="MyStyle" w:customStyle="true"> <w:name w:val="My Paragraph Style"/> <w:rPr> <w:b/> </w:rPr> </w:style>` This paragraph style specifies that it is a user-defined style using the customStyle attribute's value of `true`. An application can therefore take action on the style if it has behaviors associated with the style ID `MyStyle`. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

default (Default Style)	Specifies that this style is the default for this style type.
	
	This property is used in conjunction with the type attribute to determine the style which is applied to objects that do not explicitly declare a style. [*Example*: The paragraph style with the default attribute set is the paragraph style applied to all paragraphs which do not explicitly reference a paragraph style using the pStyle element (§17.3.1.27). *end example*]
	
	If this attribute is not specified for any style, then no properties shall be applied to objects of the specified style type. If this attribute is specified by multiple styles, then the last instance of a style with this property shall be used.
	
	[*Example*: Consider a paragraph style defined as follows:
	
	`<w:style w:type="paragraph" w:default="1" w:styleId="MyStyle" > <w:name w:val="My Paragraph Style"/> <w:rPr> <w:b/> </w:rPr> </w:style>`
	
	This paragraph style specifies that it is the default paragraph style, and therefore all paragraphs which do not explicitly reference a paragraph style must have this style applied.
	
	For example, consider the following paragraphs from the same WordprocessingML document:
	
	`<w:p> <w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pStyle w:val="Normal"/> </w:pPr> … </w:p> <w:p> … </w:p> The contents of the first paragraph must have the Normal paragraph style applied to them, while the contents of the second paragraph must have the MyStyle paragraph style applied, since it does not explicitly reference a paragraph style and therefore inherits the default. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
styleId (Style ID)	Specifies a unique identifier for the parent style definition. This identifier shall be used in multiple contexts to uniquely reference this style definition within the document. [*Example*: The following are examples of elements which reference a style via its styleId attribute value: • To reference a style from content using elements like the pStyle element (§17.3.1.27), rStyle element (§17.3.2.29), and the tblStyle element (§17.4.62) for paragraphs, runs, and tables, respectively. • To link the paragraph and character versions of a style via the link element (§17.7.4.6) • To reference the parent style for style inheritance via the basedOn element (§17.7.4.3) *end example*] If multiple style definitions each declare the same value for their styleId, then the first such instance shall keep its current identifier with all other instances being reassigned in any manner desired. This reassignment shall not require references to those style definitions to be 'repaired' in the content (i.e. some content might lose its style definition information, since the document was ill-formed). If this attribute is not specified, then a style ID can be assigned in any manner desired. [*Example*: Consider a paragraph style defined as follows: <w:style w:type="paragraph" w:styleId="MyStyle" > <w:name w:val="My Paragraph Style"/> <w:rPr> <w:b/> </w:rPr> </w:style>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	This paragraph style specifies that its style identifier must be `MyStyle` using the styleId attribute.
	
	Now consider the following paragraphs from the same WordprocessingML document:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:pStyle w:val="MyStyle"/>`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	`<w:p>`
	`  …`
	`</w:p>`
	
	The contents of the first paragraph must have the bold paragraph property applied to them because their paragraph properties specify that they must inherit the paragraph style whose styleId is `MyStyle` therefore inheriting its properties using the rules of the style hierarchy. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

type (Style Type)	Specifies the type of style definition defined by this element. WordprocessingML supports six types of style definitions:
	• Paragraph styles
	• Character styles
	• Table styles
	• Numbering styles
	• Linked styles (paragraph + character)
	• Default paragraph + character properties
	
	Each of the first four style types corresponds to a different value in this attribute, and therefore defines the style type of the current style. [*Note*: The last two style types are unique in that they are not simply a style type: a linked style is a pairing of a character and paragraph style via the link element (§17.7.4.6); and the document default properties are defined via the docDefaults element (§17.7.5.1). *end note*]
	
	If this attribute is not specified, then the default value shall be assumed to be `paragraph`.
	
	[*Example*: Consider a style defined as follows:
	
	`<w:style w:type="paragraph" … >`
	`  <w:name w:val="My Paragraph Style"/>`
	`  <w:rPr>`
	`  <w:b/>`
	`  </w:rPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</w:style> The type attribute value of paragraph specifies that this style definition creates a paragraph style. *end example*] The possible values for this attribute are defined by the ST_StyleType simple type (§17.18.83).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Style) is located in §A.1. *end note*]

17.7.4.18 styles (Style Definitions)

This element specifies all of the style information stored in the WordprocessingML document: style definitions as well as latent style information.

[*Example*: The Normal paragraph style in a word processing document can have any number of formatting properties, e.g. font face = Times New Roman; font size = 12pt; paragraph justification = left). All paragraphs which reference this paragraph style would automatically inherit these properties. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Styles) is located in §A.1. *end note*]

17.7.4.19 uiPriority (Optional User Interface Sorting Order)

This element specifies a number which can be used to sort the set of style definitions in a user interface when this document is loaded by an application and the recommended setting is specified in the stylePaneSortMethod element (§17.15.1.86). If this element is set, then this priority shall be used to sort all available styles in ascending value order.

If this element is omitted, then the style shall not have an associated priority value and shall be sorted to the end of the list of style definitions (more or less equivalent to a priority value of infinity) when the recommended sort order setting is specified.

[*Example*: Consider a style with a primary name of Comment Style that should have an associated priority value of ten. This requirement would be specified using the following WordprocessingML:

```
<w:style … w:styleId="CStyle">
  <w:name w:val="Comment Style"/>
  <w:uiPriority w:val="10"/>
  …
</w:style>
```

The uiPriority element specifies that this style definition should be sorted into the list of styles using a value of 10 when the styles are listed in recommended order using the stylePaneSortMethod element (§17.15.1.86). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.7.4.20 unhideWhenUsed (Remove Semi-Hidden Property When Style Is Used)

This element specifies whether the semiHidden property (§17.7.4.16) shall be removed when this style is used by the content of the document. If this element is set, then an application shall ensure that even if the semiHidden element is specified on a style, that this property is removed when the document is resaved if the style is referenced by any content in the document.

If this element is omitted, then the style shall not automatically lose the semi-hidden property when it is used in the document contents.

[*Example*: Consider a style with a primary name of `Test Paragraph Style` that should not be displayed in the main user interface until it is used. This requirement would be specified using the following WordprocessingML:

```
<w:style … w:styleId="TestStyle">
  <w:name w:val="Test Paragraph Style"/>
  <w:semiHidden/>
  <w:unhideWhenUsed/>
  …
</w:style>
```

The unhideWhenUsed element specifies that this style definition should not be displayed in any main user interface associated with an application which processes this document until it is referenced by document content. If a paragraph was added to the document which referenced this style:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="TestStyle"/>
  </w:pPr>
  …
</w:p>
```

This style is now referenced by the document's contents and would have the semiHidden element removed on save. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.7.5 Document Defaults

The first formatting information which is applied to all regions of text in a WordprocessingML document when that document is displayed is the document defaults. The document defaults specify the default set of properties which shall be inherited by every paragraph and run of text within all stories of the current WordprocessingML document. If no other formatting information was referenced by that text, these properties would solely define the formatting of the resulting text.

[*Example*: Consider the following fragment from the main document part of a WordprocessingML document:

```
<w:body>
  <w:p>
    <w:r>
      <w:t>Hello, world</w:t>
    </w:r>
  </w:p>
</w:body>
```

This paragraph and run of text both specify no formatting information (i.e. the paragraph and run possess neither a pPr element nor an rPr element respectively). Therefore, the only formatting applied to this text shall be the formatting in the document defaults which is applied to all paragraphs and runs in the document.

Note that this does not imply that these properties are only applied to text with no formatting - they are rather applied to all text before all other formatting (and in this case, there is no other formatting). *end example*]

17.7.5.1 docDefaults (Document Default Paragraph and Run Properties)

This element specifies the set of default paragraph and run properties which shall be applied to every paragraph and run in the current WordprocessingML document. These properties are applied first in the style hierarchy; therefore they are superseded by any further conflicting formatting, but apply if no further formatting is present.

If this element is omitted, then the document defaults shall be application-defined by the hosting application.

[*Example*: Consider the following definition for the document defaults for a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:docDefaults>
  <w:rPrDefault>
    <w:rPr>
      <w:b/>
    </w:rPr>
  </w:rPrDefault>
  <w:pPrDefault>
    <w:pPr>
      <w:jc w:val="center"/>
    </w:pPr>
  </w:pPrDefault>
</w:docDefaults>
```

The child elements of docDefaults specify a default paragraph property of centered text and a default run property of bold text. Applying this formatting to the following fragment from the main document part of the same document:

```
<w:body>
  <w:p>
    <w:r>
      <w:t>Hello, world</w:t>
    </w:r>
  </w:p>
</w:body>
```

This paragraph contains no formatting properties, therefore, using the style hierarchy the document default paragraph and run properties are applied as specified within the docDefaults element and the resulting paragraph is centered as specified in the jc element (§17.3.1.13) as well as bold as specified via the b element (§17.3.2.1). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocDefaults) is located in §A.1. *end note*]

17.7.5.2 pPr (Paragraph Properties)

This element specifies the set of paragraph properties which comprise the default paragraph properties for the current WordprocessingML document. [*Rationale*: The reason that a pPr element is present within the pPrDefault element is to allow for easy repurposing of any set of paragraph properties within a WordprocessingML document - since the paragraph properties are always child elements of a single pPr element, that element can simply be relocated in its entirety to the desired new location without additional modifications. *end rationale*]

If this element is omitted, then the default paragraph properties for the current document are non-existent (i.e. there are no default paragraph properties, and the defaults are therefore application-defined).

[*Example*: Consider the following definition for the document defaults for a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:docDefaults>
  <w:pPrDefault>
    <w:pPr>
      <w:jc w:val="center"/>
    </w:pPr>
  </w:pPrDefault>
  …
</w:docDefaults>
```

The pPr element as a child of the pPrDefault element contains the set of default paragraph properties for this document - in this case, a justification value of `center`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrGeneral) is located in §A.1. *end note*]

17.7.5.3 pPrDefault (Default Paragraph Properties)

This element specifies the presence of a set of default paragraph properties for the current document. The actual paragraph properties are stored within the pPr child element of the current element.

If this element is omitted, then the default paragraph properties for the current document are non-existent (i.e. there are no default paragraph properties in the document, and the defaults are therefore application-defined).

[*Example*: Consider the following definition for the document defaults for a WordprocessingML document:

```
<w:docDefaults>
  <w:pPrDefault>
    <w:pPr>
      <w:jc w:val="center"/>
    </w:pPr>
  </w:pPrDefault>
  …
</w:docDefaults>
```

The pPrDefault element is a container for the set of default paragraph properties for this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrDefault) is located in §A.1. *end note*]

17.7.5.4 rPr (Run Properties)

This element specifies the set of run properties which comprise the default run properties for the current WordprocessingML document. [*Rationale*: The reason that an rPr element is present within the rPrDefault element is to allow for easy repurposing of any set of run properties within a WordprocessingML document - since the run properties are always child elements of a single rPr element, that element can simply be relocated in its entirety to the desired new location without additional modifications. *end rationale*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then the default run properties for the current document are non-existent (i.e. there are no default run properties, and the defaults are therefore application-defined).

[*Example*: Consider the following definition for the document defaults for a WordprocessingML document:

```
<w:docDefaults>
  …
  <w:rPrDefault>
    <w:rPr>
      <w:b/>
    </w:rPr>
  </w:rPrDefault>
</w:docDefaults>
```

The rPr element as a child of the rPrDefault element contains the set of default run properties for this document - in this case, bold text. *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language.*end note*]

17.7.5.5 rPrDefault (Default Run Properties)

This element specifies the presence of a set of default run properties for the current document. The actual run properties are stored within the rPr child element of the current element.

If this element is omitted, then the default run properties for the current document are non-existent (i.e. there are no default run properties in the document, and the defaults are therefore application-defined).

[*Example*: Consider the following definition for the document defaults for a WordprocessingML document:

```
<w:docDefaults>
  …
  <w:rPrDefault>
    <w:rPr>
      <w:b/>
    </w:rPr>
  </w:rPrDefault>
</w:docDefaults>
```

The rPrDefault element is a container for the set of default run properties for this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RPrDefault) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.7.6 Table Styles

Table styles are style definitions which apply to the contents of zero or more tables within a document. This definition can imply that the style can only define table properties (properties which apply to the table and its constituent rows and cells), however a table style can also define paragraph properties (properties which apply to the positioning and appearance of paragraphs) as well as character properties (properties which apply to runs) for all of the paragraphs and runs within the specified table in the document.

Table styles can only be referenced by tables within a document, and they shall be referenced by the tblStyle element (§17.4.62) within a table's table properties.

As discussed above, table styles can specify all of the properties that can be applied to a table, as well as paragraph and character properties for the table's contents. However, unlike other style definitions, table styles allow for the definition of conditional formats for different regions of the table.

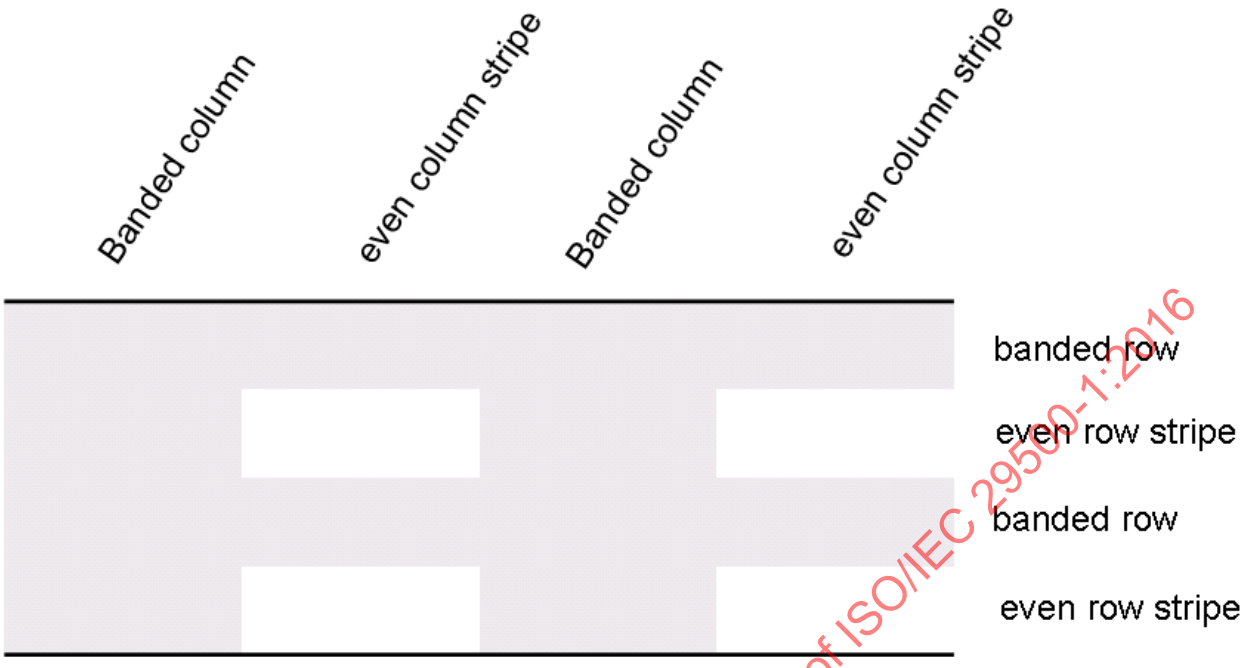
These table conditional formats are applied to different regions of the table as follows:

Top left cell	Header row	Top right cell
First column	Table body	Last column
Bottom left cell	Footer row	Bottom right cell

All rows in the table can also have conditional formatting on an alternating row/column basis as well as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



When specified, these conditional formats shall be applied in the following order (therefore subsequent formats override properties on previous formats):

- Whole table
- Banded columns, even column banding
- Banded rows, even row banding
- First row, last row
- First column, last column
- Top left, top right, bottom left, bottom right

[*Example*: Consider a table style `Test Table Style` defined as follows:

- All cells with 1pt table borders on all sides
- 0.1” cell margins on left and right of cells
- 0” cell margins on top and bottom of cells

As well as header row specific formatting of

- Red shading
- Bold text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:style w:type="table" w:styleId="TestTableStyle">
  <w:name w:val="Test Table Style"/>
  <w:basedOn w:val="TableNormal"/>
  <w:uiPriority w:val="99"/>
  <w:rsid w:val="00340CC4"/>
  <w:tblPr>
    <w:tblBorders>
      <w:top w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:start w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:bottom w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:end w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:insideH w:val="single" w:sz="4" w:space="0" w:color="auto"/>
      <w:insideV w:val="single" w:sz="4" w:space="0" w:color="auto"/>
    </w:tblBorders>
    <w:tblCellMar>
      <w:top w:w="0" w:type="dxa"/>
      <w:start w:w="108" w:type="dxa"/>
      <w:bottom w:w="0" w:type="dxa"/>
      <w:end w:w="108" w:type="dxa"/>
    </w:tblCellMar>
  </w:tblPr>
  <w:tblStylePr w:type="firstRow">
    <w:rPr>
      <w:b/>
    </w:rPr>
    <w:tcPr>
      <w:shd w:val="clear" w:color="auto" w:fill="ED1C24"/>
    </w:tcPr>
  </w:tblStylePr>
</w:style>
```

The tblPr element holds the formatting which is applied to the entire table, and the tblStylePr element with a type attribute value of `firstRow` holds the formatting for the first table row, specifically the bold run property and red cell shading. *end example*]

An individual instance of a table defines an association with a table style using the tblStyle element in the table's properties (tblPr), as discussed above. However, individual tables can choose whether to apply the following aspects of the table's conditional formats individually:

- First row
- Last row
- First column
- Last column
- Row banding

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Column banding

The use or omission of conditional formats shall be specified using the tblLook element, which contains a a number of attributes that indicate which properties are applied and omitted.

[*Example*: Consider two tables using the table style `Style2`; one which specifies that it should only use the header row and footer row conditional formatting properties from the table style, and the other which specifies that it should use the header row, footer row, and banded row conditional formatting:

```
<w:tbl>
  <w:tblPr>
    <w:tblStyle w:val="Style2"/>
    <w:tblW w:w="0" w:type="auto"/>
    <w:tblLook w:firstRow="true" w:lastRow="true"
      w:noHBand="true" w:noVBand="true" />
  </w:tblPr>
  …
</w:tbl>
…
<w:tbl>
  <w:tblPr>
    <w:tblStyle w:val="Style2"/>
    <w:tblW w:w="0" w:type="auto"/>
    <w:tblLook w:firstRow="true" w:lastRow="true"
      w:noVBand="true" />
  </w:tblPr>
  …
</w:tbl>
```

The tables each specify the appropriate set of conditional formats using the tblLook element, as seen by the identical table styles in the tblStyle element, and different tblLook values. *end example*]

17.7.6.1 pPr (Table Style Conditional Formatting Paragraph Properties)

This element specifies the set of paragraph properties which shall be applied to all paragraphs within a table which match the conditional formatting type specified on the parent tblStylePr element. These properties are applied in the order specified via the style hierarchy.

[*Example*: Consider a table style which contains conditional formatting for its `firstRow`, defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:pPr>
      <w:jc w:val="center"/>
    </w:pPr>
    …
  </w:tblStylePr>
</w:style>
```

The pPr element specified within the tblStylePr element specifies the set of paragraph properties which must be applied to all parts of the table which meet the criteria specified by the type value of `firstRow` - all of the header rows of the table. In this example, the single paragraph property applied is an alignment value of center via the jc element (§17.3.1.13). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrGeneral) is located in §A.1. *end note*]

17.7.6.2 rPr (Table Style Conditional Formatting Run Properties)

This element specifies the set of run properties which shall be applied to all runs within a table which match the conditional formatting type specified on the parent tblStylePr element. These properties are applied in the order specified via the style hierarchy.

[*Example*: Consider a table style which contains conditional formatting for its `firstRow`, defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:rPr>
      <w:i/>
    </w:rPr>
    …
  </w:tblStylePr>
</w:style>
```

The rPr element specified within the tblStylePr element specifies the set of run properties which must be applied to all parts of the table which meet the criteria specified by the type value of `firstRow` - all of the header rows of the table. In this example, the single run property applied is italics via the i element (§17.3.2.16). *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language.*end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.7.6.3 tblPr (Table Style Conditional Formatting Table Properties)

This element specifies the set of table properties which shall be applied to all regions within a table which match the conditional formatting type specified on the parent tblStylePr element. These properties are applied in the order specified via the style hierarchy.

If the current conditional formatting type does not consist of one or more full table rows, then table properties which cannot be applied to a single cell or column [*Example*: Table justification. *end example*] can be ignored.

[*Example*: Consider a table style which contains conditional formatting for its `firstRow`, defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:tblPr>
      <w:tblCellSpacing w:w="29" w:type="dxa"/>
    </w:tblPr>
    …
  </w:tblStylePr>
</w:style>
```

The tblPr element specified within the tblStylePr element specifies the set of table properties which must be applied to all parts of the table which meet the criteria specified by the type value of `firstRow` - all of the header rows of the table. In this example, the single table property applied is a default table cell spacing value of 0.02 inches via the tblCellSpacing element (§17.4.45). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrBase) is located in §A.1. *end note*]

17.7.6.4 tblPr (Style Table Properties)

This element specifies the set of table properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

[*Example*: Consider a table style defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  <w:tblPr>
    <w:tblCellSpacing w:w="15" w:type="dxa"/>
  </w:tblPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    <w:tblStylePr w:type="firstRow">-
      <w:tblPr>
        <w:tblCellSpacing w:w="29" w:type="dxa"/>
      </w:tblPr>
      …
    </w:tblStylePr>
  </w:style>
```

The tblPr element specified within the style element specifies the set of table properties which must be applied to all parts of the table. In this example, the single table property applied is a default table cell spacing value of 0.01 inches via the tblCellSpacing element (§17.4.45). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrBase) is located in §A.1. *end note*]

17.7.6.5 tblStyleColBandSize (Number of Columns in Column Band)

This element specifies the number of columns which shall comprise each a table style column band for this table style. This element determines how many columns constitute each of the column bands for the current table, allowing column band formatting to be applied to groups of columns (rather than just single alternating columns) when the table is formatted.

If this element is omitted, then the default number of columns in a single column band shall be assumed to be 1.

[*Example*: Consider a table style defined as follows:

```
  <w:style w:type="table" w:styleId="exampleTableStyle">
    <w:tblPr>
      <w:tblStyleRowBandSize w:val="3" />
      <w:tblStyleColBandSize w:val="2" />
    </w:tblPr>
    …
  </w:style>
```

The tblStyleColBandSize element specifies that the width of each column band must be 2 columns - therefore band1Vert column banding conditional formatting must be applied to columns 1 and 2, 5 and 6, etc. in the table. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

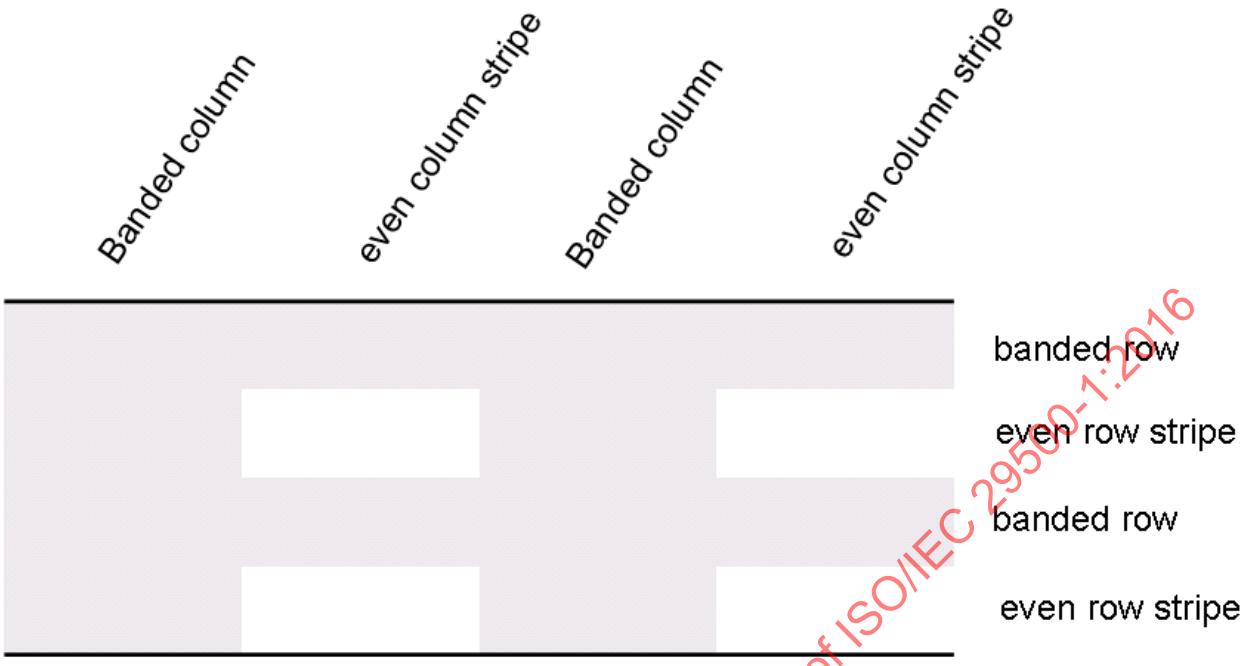
17.7.6.6 tblStylePr (Style Conditional Table Formatting Properties)

This element specifies a set of formatting properties which shall be conditionally applied to the parts of a table which match the requirement specified on the type attribute. These table conditional formats are applied to different regions of the table as follows:

Top left cell	Header row	Top right cell
First column	Table body	Last column
Bottom left cell	Footer row	Bottom right cell

All rows in the table can also have conditional formatting on an alternating row/column basis as well as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



When specified, these conditional formats shall be applied in the following order (therefore subsequent formats override properties on previous formats):

- Whole table
- Banded columns, even column banding
- Banded rows, even row banding
- First row, last row
- First column, last column
- Top left, top right, bottom left, bottom right

[*Example*: Consider a table style which contains conditional formatting, defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:tblPr>
      <w:tblCellSpacing w:w="29" w:type="dxa"/>
    </w:tblPr>
    …
  </w:tblStylePr>
</w:style>
```

The tblStylePr element specifies a set of table properties which must be conditionally applied to all parts of the table which meet the criteria specified by the type attribute (in this case, all heading rows for the current table). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

type (Table Style Conditional Formatting Type)	Specifies the section of the table to which the current conditional formatting properties shall be applied.
	
	[*Example*: Consider a table style which contains conditional formatting, defined as follows:
	
	`<w:style w:type="table" …>`
	`  …`
	`  <w:tblStylePr w:type="lastRow">`
	`  …`
	`  </w:tblStylePr>`
	`</w:style>`
	
	The type attribute value of `lastRow` specifies that this set of conditional formatting properties must be applied to the last row of the table only. *end example*]
	
	The possible values for this attribute are defined by the ST_TblStyleOverrideType simple type (§17.18.89).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblStylePr) is located in §A.1. *end note*]

17.7.6.7 tblStyleRowBandSize (Number of Rows in Row Band)

This element specifies the number of rows which shall comprise each a table style row band for this table style. This element determines how many rows constitute each of the row bands for the current table, allowing row band formatting to be applied to groups of rows (rather than just single alternating rows) when the table is formatted.

If this element is omitted, then the default number of rows in a single row band shall be assumed to be `1`.

[*Example*: Consider a table style defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  <w:tblPr>
    <w:tblStyleRowBandSize w:val="3" />
    <w:tblStyleColBandSize w:val="2" />
  </w:tblPr>
  …
</w:style>
```

The tblStyleRowBandSize element specifies that the width of each row band must be `3` columns - therefore `band1Horiz` row banding conditional formatting must be applied to row 1 through 3, 7 through 9, etc. in the table. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.7.6.8 tcPr (Table Style Conditional Formatting Table Cell Properties)

This element specifies the set of table cell properties which shall be applied to all regions within a table which match the conditional formatting type specified on the parent tblStylePr element. These properties are applied in the order specified via the style hierarchy.

[*Example*: Consider a table style which contains conditional formatting for its `firstRow`, defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:tcPr>
      <w:tcBorders>
        <w:top w:val="nil" />
        <w:start w:val="nil" />
        <w:bottom w:val="nil" />
        <w:end w:val="nil" />
        <w:insideH w:val="nil" />
        <w:insideV w:val="nil" />
      </w:tcBorders>
    </w:tcPr>
    …
  </w:tblStylePr>
</w:style>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The tcPr element specified within the tblStylePr element specifies the set of table cell properties which must be applied to all parts of the table which meet the criteria specified by the type value of `firstRow` - all of the header rows of the table. In this example, the single table cell property applied is a set of table cell borders via the tcBorders element (§17.4.66). In this case, these cell borders simply reset any previous cell borders to `nil`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcPr) is located in §A.1. *end note*]

17.7.6.9 tcPr (Style Table Cell Properties)

This element specifies the set of table cell properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

[*Example*: Consider a table style defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  <w:tcPr>
    <w:tcFitText/>
  </w:tcPr>
</w:style>
```

The tcPr element specified within the style element specifies the set of table cell properties which must be applied to all parts of the table. In this example, the single table cell property applied is the fit text setting via the tcFitText element (§17.4.67). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcPr) is located in §A.1. *end note*]

17.7.6.10 trPr (Table Style Conditional Formatting Table Row Properties)

This element specifies the set of table row properties which shall be applied to all rows within a table which match the conditional formatting type specified on the parent tblStylePr element. These properties are applied in the order specified via the style hierarchy.

[*Example*: Consider a table style which contains conditional formatting for its `firstRow`, defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  …
  <w:tblStylePr w:type="firstRow">
    <w:trPr>
      <w:tblHeader/>
      <w:cantSplit/>
    </w:trPr>
    …
  </w:tblStylePr>
</w:style>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The trPr element specified within the tblStylePr element specifies the set of table row properties which must be applied to all rows of the table which meet the criteria specified by the type value of `firstRow` - all of the header rows of the table. In this example, the table row properties applied are the fact that these rows must be repeated on each page via the tblHeader element (§17.4.49) and the fact that these rows must not be split across pages using the cantSplit element (§17.4.6). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrPr) is located in §A.1. *end note*]

17.7.6.11 trPr (Style Table Row Properties)

This element specifies the set of table row properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

[*Example*: Consider a table style defined as follows:

```
<w:style w:type="table" w:styleId="exampleTableStyle">
  <w:trPr>
    <w:jc w:val="center"/>
  </w:trPr>
</w:style>
```

The trPr element specified within the style element specifies the set of table row properties which must be applied to all parts of the table. In this example, the single table row property applied is the alignment setting of center via the jc element (§17.4.27). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrPr) is located in §A.1. *end note*]

17.7.7 Numbering Styles

Numbering styles are style definitions which specify common style properties for a multi-level numbering format within a document. This means that a numbering style defines only a single paragraph property: a reference to a numbering definition stored in the document's numbering part, using the numPr element.

Unlike paragraph and character styles, numbering styles are never directly referenced by content in the document – instead, an abstract numbering definition (covered in the numbering section) specifies that it is actually the underlying numbering information for a numbering style.

[*Example*: Consider a numbering style "Test Numbering Style":

```
<w:style w:type="numbering" w:styleId="TestNumberingStyle">
  <w:name w:val="Test Numbering Style" />
  <w:uiPriority w:val="99" />
  <w:rsid w:val="0045009F" />
  <w:pPr>
    <w:numPr>
      <w:numId w:val="1" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      </w:numPr>
    </w:pPr>
  </w:style>
```

The only information specified in the numbering style definition is a reference to the numbering definition for the numbering information which is defined by this numbering style. *end example*]

17.7.8 Paragraph Styles

Paragraph styles are styles which apply to the contents of an entire paragraph as well as the paragraph mark. This definition implies that the style can define both character properties (properties which apply to text within the document) as well as paragraph properties (properties which apply to the positioning and appearance of the paragraph). Paragraph styles cannot be referenced by runs within a document; they shall be referenced by the pStyle element (§17.3.1.27) within a paragraph's paragraph properties element.

A paragraph style has three defining style type-specific characteristics:

- The type attribute on the style has a value of `paragraph`, which indicates that the following style definition is a paragraph style.
- The next element defines an editing behavior which supplies the paragraph style to be automatically applied to the next paragraph when ENTER is pressed at the end of a paragraph of this style.
- The style specifies both paragraph-level and character-level properties using the pPr and rPr elements, respectively. In this case, the run properties are the set of properties applied to each run in the paragraph.

The paragraph style is then applied to paragraphs by referencing the styleId attribute value for this style in the paragraph properties' pStyle element.

[*Example:* Consider a paragraph style titled "Test Paragraph Style" which defines; font = Algerian, font size = 20; font color = red; paragraph spacing = double; paragraph indent = 1" (first line only). The resulting style definition would be:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:style w:type="paragraph" w:styleId="TestParagraphStyle">
  <w:name w:val="Test Paragraph Style"/>
  <w:qFormat/>
  <w:rsid w:val="00F85845"/>
  <w:pPr>
    <w:spacing w:line="480" w:lineRule="auto"/>
    <w:ind w:firstLine="1440"/>
  </w:pPr>
  <w:rPr>
    <w:rFonts w:ascii="Algerian" w:hAnsi="Algerian"/>
    <w:color w:val="ED1C24"/>
    <w:sz w:val="40"/>
  </w:rPr>
</w:style>
```

Notice that the character properties for the style are under the rPr element, and the paragraph properties are under the pPr element.

The document content for a paragraph of this style would be:

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="TestParagraphStyle"/>
  </w:pPr>
  <w:r>
    <w:t xml:space="preserve">Here is some fancy Text</w:t>
  </w:r>
</w:p>
```

The pStyle element links the paragraph with the style definition. *end example*]

17.7.8.1 Numbering in Paragraph Styles

When a paragraph style references a numbering definition and level which shall also be applied, that reference shall be done in a way slightly different from the typical numbering reference as follows:

- When a numbering reference is created as direct formatting, that reference consists of a reference to the numbering definition instance + a numbering level

[*Example*: Consider a numbered paragraph in a WordprocessingML document whose numbering is a result of direct formatting (formatting not from a style). This numbered paragraph might be represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Level one</w:t>
  </w:r>
</w:p>
```

The numPr element contains two pieces of information:

- The numId element (the numbering definition instance referenced)
- The ilvl element (the level within that numbering definition)

end example]

- When numbering is done as part of a paragraph style, that reference consists of a reference to the numbering definition only. The numbering definition then in turn has a reference to the paragraph style on the level which shall be associated with this style

[*Example*: Consider a numbered paragraph in a WordprocessingML document whose numbering is a result of a paragraph style. This numbered paragraph might be represented using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:pStyle w:val="TestParagraphStyle"/>
  </w:pPr>
  <w:r>
    <w:t>Level one</w:t>
  </w:r>
</w:p>
```

The paragraph references the style via its styleId attribute, which itself looks like this:

```
<w:style w:styleId="TestParagraphStyle" … >
  <w:pPr>
    <w:numPr>
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
</w:style>
```

The numPr element contains one piece of information:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- The numId element (the numbering definition instance referenced)

Obviously, this is insufficient to apply the numbering since we need to know which level to apply, so this information is specified on the appropriate level using the pStyle element:

```
<w:abstractNum w:abstractNumId="1">
  …
  <w:lvl w:ilvl="0">
    …
    <w:pStyle w:val="TestParagraphStyle" />
    <w:pPr>
      <w:tabs>
        <w:tab w:val="num" w:pos="720" />
      </w:tabs>
      <w:ind w:start="720" w:hanging="360" />
    </w:pPr>
    …
  </w:lvl>
</w:abstractNum>
```

In this case, level `0` of the underlying abstract numbering definition specifies that it is associated with paragraph style `TestParagraphStyle`, so this level of the numbering must be applied along with the paragraph style. *end example*]

When numbering is referenced by a paragraph style, its properties shall be applied before the style's properties (the style's paragraph properties shall override the numbering level's paragraph properties).

17.7.8.2 pPr (Style Paragraph Properties)

This element specifies the set of paragraph properties which shall be applied to the paragraph.

[*Example*: Consider a paragraph style defined as follows:

```
<w:style w:type="paragraph" w:styleId="TestParaStyle">
  <w:pPr>
    <w:keepLines/>
  </w:pPr>
</w:style>
```

The pPr element specified within the style element specifies the set of paragraph properties which must be applied to the referencing paragraph. In this example, the single paragraph property applied is the fact that the paragraph must be displayed on a single page via the keepLines element (§17.3.1.14). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrGeneral) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.7.9 Run (Character) Styles

Character styles are styles which apply to the contents of one or more runs of text within a document's contents. This definition implies that the style can only define character properties (properties which apply to text within a paragraph) because it cannot be applied to paragraphs. Character styles can only be referenced by runs within a document, and they shall be referenced by the rStyle element within a run's run properties element.

A character style has two defining style type-specific characteristics:

- The type attribute on the style has a value of `character`, which indicates that the following style definition is a character style.
- The style specifies only character-level properties using the rPr element. In this case, the run properties are the set of properties applied to each run which is of this style.

The character style is then applied to runs by referencing the styleId attribute value for this style in the run properties' rStyle element.

[*Example:* Consider a character style titled "Test Character Style" which defines; font = Courier New, font color = yellow; underline. The resulting style definition would be:

```
<w:style w:type="character" w:styleId="TestCharacterStyle">
  <w:name w:val="Test Character Style"/>
  <w:uiPriority w:val="99"/>
  <w:qFormat/>
  <w:rsid w:val="00E77BF0"/>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:hAnsi="Courier New"/>
    <w:color w:val="FFF200"/>
    <w:u w:val="single"/>
  </w:rPr>
</w:style>
```

Notice that the character properties applied using this style are under the rPr element. The document content for a paragraph with a run of this style would be:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">The following text is in the </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="TestCharacterStyle"/>
    </w:rPr>
    <w:t>character style</w:t>
  </w:r>
  <w:r>
    <w:t>.</w:t>
  </w:r>
</w:p>
```

The rStyle element in the second run links that run with the style definition, inheriting the formatting properties for that run. *end example*]

17.7.9.1 rPr (Run Properties)

This element specifies the set of run properties which shall be applied to the run.

[*Example*: Consider a character style defined as follows:

```
<w:style w:type="character" w:styleId="TestCharStyle">
  <w:rPr>
    <w:dstrike/>
  </w:rPr>
</w:style>
```

The rPr element specified within the style element specifies the set of run properties which must be applied to the referencing run. In this example, the single run property applied is the fact that the paragraph must be displayed with double strikethrough via the dstrike element (§17.3.2.9). *end example*]

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language.*end note*]

17.8 Fonts

The next component of a WordprocessingML document is storing information about the fonts used in the document. WordprocessingML stores two pieces of information about fonts:

- (optionally) Information about the font to enable font substitution. *Font substitution* is a process by which an application, when it cannot locate a specific font, attempts to locate the closest possible match to the intended appearance of the font

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- (optionally) One or more embedded forms of the font for use on systems which do not have access to the font. When fonts are embedded, they are obfuscated to ensure that they are only used to view the contents of the document in which they are embedded, and for no other purpose.

[*Example*: Consider the following information stored in a document's font table part:

```
<w:fonts>
  <w:font w:name="Times New Roman">
    <w:panose1 w:val="02020603050405020304" />
    <w:charset w:val="00" />
    <w:family w:val="roman" />
    <w:pitch w:val="variable" />
    <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008"
w:usb3="00000000" w:csb0="000001FF" w:csb1="00000000" />
    <w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-
89EF99B2457E}" />
  </w:font>
</w:fonts>
```

The font table contains information about the `Times New Roman` font, specifically, information used to locate a substitute font when it is not available and a relationship to the embedded form of the regular form of the font. *end example*]

17.8.1 Font Embedding

Within a WordprocessingML document, *font embedding* refers to a process in which the some or all of the fonts used in the current document are included in that document such that it can be guaranteed that they are available for use when the document is subsequently opened.

Embedded fonts are stored in an Embedded Font part within the package.

When a font is embedded within a WordprocessingML document, it shall be obfuscated to prevent it from being used outside of this document. This obfuscation shall be done using the following algorithm:

- Generate a GUID, which is used and stored as the obfuscation key
- Reverse the order of the bytes in the GUID (i.e. Big Endian ordering)
- XOR the value with the first 32 bytes of the binary: once against 0-15, once against 16-31
- Store the resulting file in the document, and store the obfuscation key in the fontKey attribute

[*Example*: Consider a font to be embedded whose first 32 bytes are as follows:

00	01	00	00	00	12	01	00	00	04	00	20	44	53	49	47
A3	0E	BF	F3	00	01	36	8C	00	00	14	DC	4C	54	53	48

To obfuscate this font for storage:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Generate a GUID (e.g. `001B70DC-AA60-4AD5-90EC-18A0948E1EAE`)
- Reverse its order (e.g. `AE1E8E94-A018-EC90-D54A-60AADC701B00`)
- XOR the GUID with the first and second 16 bytes

The resulting 32 bytes would be:

AE	1F	8E	94	A0	0A	ED	90	D5	4E	60	8A	98	23	52	47
0D	10	31	67	A0	19	DA	1C	D5	4A	74	76	90	24	48	48

end example]

To retrieve an obfuscated font for viewing the content of this document only, repeat the procedure above to retrieve the original font.

17.8.2 Font Substitution

The rFonts element (§17.3.2.26) references the font which is applied to each run of text within a WordprocessingML document. However, based on the availability of these fonts (for example, the use of a custom font), an application might not be able to locate the specified font. The process of finding a suitable alternative font is known as *font substitution*.

The exact algorithm which is used for font substitution is highly dependent on the characteristics which are most desirable when performing the substitution: similar appearance of each glyph (to maximize visual familiarity), similar physical characteristics (to minimize changes in line height and breaking), etc. ISO/IEC 29500 recommends that applications looking for the closest match to the following pieces of information (in descending priority) in order to determine a suitable alternative font; however, applications are free to apply more sophisticated logic in its place:

- sig (§17.8.3.16)
- charset (§17.8.3.2)
- panose1 (§17.8.3.13)
- pitch (§17.8.3.14)
- family (§17.8.3.9)
- altName (§17.8.3.1)
- notTrueType (§17.8.3.12)

17.8.3 Elements

The following elements comprise the content of the font table:

17.8.3.1 altName (Alternate Names for Font)

This element specifies a set of alternative names which can be used to locate the font specified by the parent element. This set of alternative names is stored in a comma-delimited list, with all adjacent commas ignored (i.e. a value of `Name A, Name B` is equivalent to `Name A,,,,,,,, Name B`).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

When an application cannot locate a font using the primary name stored on the font attribute of the font element (§17.8.3.10), it should use each alternate name in term to attempt to locate the font, and use the first font for which is locates a match.

Font names stored using this element shall be specified in the encoding specified by the Fonts part in its XML declaration; the name of the font will be interpreted by the XML parser. [*Note:* UTF-8 is not supported for font names within OpenType fonts, and is not always supported for file names in file systems. For example, older Japanese versions of Microsoft Windows use Windows-31J for filenames. To use extant font names in the values of this attribute, they should be converted to the character encoding appropriate for this standard and copying raw byte sequences from font files should be avoided. *end note*]

If this element is omitted, then no alternate names are present for the parent font.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="SimSun">
  <w:altName w:val="Arial Unicode MS" />
  …
</w:font>
```

The altName element specifies that when no font with a name of `SimSun` (the primary font name) can be located, that applications should attempt to locate a font with the name `Arial Unicode MS` before doing substitution based on the font metrics. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.8.3.2 charset (Character Set Supported By Font)

This element specifies the character set which is supported by the parent font. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is not present, then the character set for this font shall be assumed to be the ISO/IEC 8859-1 character set.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="SimSun">
  <w:charset w:characterSet="GBK" />
  …
</w:font>
```

The charset element specifies via its characterSet attribute value of GBK that this font uses the GBK character set. *end example*]

Attributes	Description
characterSet (IANA Name of Character Set)	Name of the character set associated with the font. The values allowed by this attribute are defined by the names and aliases listed in the IANA registration table. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Charset) is located in §A.1. *end note*]

17.8.3.3 embedBold (Bold Style Font Style Embedding)

This element specifies information about the embedded font storage for the bold form of a font, when it is embedded. This form is used when bold is applied to a text run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then no bold form of the font is stored in the document. The relationship targeted by the id attribute shall be of the embedded font type, or the document shall be considered to be non-conformant.

[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  …
  <w:embedBold r:id="rId10" />
</w:font>
```

The embedBold element specifies that the embedded font targeted with the relationship with ID `rId10` can be used to retrieve the bold form of the embedded Arial font. *end example*]

Attributes	Description

fontKey (Embedded Font Obfuscation Key)	Specifies the key which was used to obfuscate this embedded font. This key can be used to retrieve the embedded font for the purposes of viewing this WordprocessingML document only, using the algorithm described in §17.8.1.
	
	If this attribute is omitted, then no key is provided for this font.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" />`
	`</w:font>`
	
	The fontKey attribute has a value of `{302EE813-EB4A-4642-A93A-89EF99B2457E}`, therefore the embedded Arial font targeted with the relationship with ID `rId10` can be retrieved if needed by using this key and the algorithm above. *end example*]
	
	The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

subsetted (Embedded Font Is Subsetted)	Specifies that the embedded font targeted by the id attribute has been subsetted. *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.
	
	If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetted.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:subsetted="true" />`
	`</w:font>`
	
	The subsetted attribute has a value of `true`, therefore the embedded Arial font targeted with the relationship with ID `rId10` must be treated as a subsetted font. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontRel) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.8.3.4 embedBoldItalic (Bold Italic Font Style Embedding)

This element specifies information about the embedded font storage for the bold italic form of a font, when it is embedded. This form is used when bold and italics are applied to a text run.

If this element is omitted, then no bold italic form of the font is stored in the document.

[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  …
  <w:embedBoldItalic r:id="rId11" />
</w:font>
```

The embedBoldItalic element specifies that the embedded font targeted with the relationship with ID `rId11` can be used to retrieve the bold italic form of the embedded Arial font. *end example*]

Attributes	Description

fontKey (Embedded Font Obfuscation Key)	Specifies the key which was used to obfuscate this embedded font. This key can be used to retrieve the embedded font for the purposes of viewing this WordprocessingML document only, using the algorithm described in §17.8.1.
	
	If this attribute is omitted, then no key is provided for this font.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" />`
	`</w:font>`
	
	The fontKey attribute has a value of `{302EE813-EB4A-4642-A93A-89EF99B2457E}`, therefore the embedded Arial font targeted with the relationship with ID `rId10` can be retrieved if needed by using this key and the algorithm above. *end example*]
	
	The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationshi	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/customXml • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
ps	• http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

subsetted (Embedded Font Is Subsetted)	Specifies that the embedded font targeted by the id attribute has been subsetted. *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.
	
	If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetted.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:subsetted="true" />`
	`</w:font>`
	
	The subsetted attribute has a value of `true`, therefore the embedded Arial font targeted with the relationship with ID `rId10` must be treated as a subsetted font. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontRel) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.8.3.5 embedItalic (Italic Font Style Embedding)

This element specifies information about the embedded font storage for the italic form of a font, when it is embedded. This form is used when italics are applied to a text run.

If this element is omitted, then no italic form of the font is stored in the document.

[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  …
  <w:embedItalic r:id="rId12" />
</w:font>
```

The embedItalic element specifies that the embedded font targeted with the relationship with ID `rId12` can be used to retrieve the italic form of the embedded Arial font. *end example*]

Attributes	Description

fontKey (Embedded Font Obfuscation Key)	Specifies the key which was used to obfuscate this embedded font. This key can be used to retrieve the embedded font for the purposes of viewing this WordprocessingML document only, using the algorithm described in §17.8.1.
	
	If this attribute is omitted, then no key is provided for this font.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" />`
	`</w:font>`
	
	The fontKey attribute has a value of `{302EE813-EB4A-4642-A93A-89EF99B2457E}`, therefore the embedded Arial font targeted with the relationship with ID `rId10` can be retrieved if needed by using this key and the algorithm above. *end example*]
	
	The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationshi	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
ps	footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

subsetted (Embedded Font Is Subsetted)	Specifies that the embedded font targeted by the id attribute has been subsetted. *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.
	
	If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetted.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:subsetted="true" />`
	`</w:font>`
	
	The subsetted attribute has a value of `true`, therefore the embedded Arial font targeted with the relationship with ID `rId10` must be treated as a subsetted font. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontRel) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.8.3.6 embedRegular (Regular Font Style Embedding)

This element specifies information about the embedded font storage for the regular form of a font, when it is embedded. This form is used when neither bold nor italics is applied to a text run.

If this element is omitted, then no regular form of the font is stored in the document.

[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  …
  <w:embedRegular r:id="rId13" />
</w:font>
```

The embedRegular element specifies that the embedded font targeted with the relationship with ID `rId13` can be used to retrieve the regular form of the embedded Arial font. *end example*]

Attributes	Description

fontKey (Embedded Font Obfuscation Key)	Specifies the key which was used to obfuscate this embedded font. This key can be used to retrieve the embedded font for the purposes of viewing this WordprocessingML document only, using the algorithm described in §17.8.1.
	
	If this attribute is omitted, then no key is provided for this font.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" />`
	`</w:font>`
	
	The fontKey attribute has a value of `{302EE813-EB4A-4642-A93A-89EF99B2457E}`, therefore the embedded Arial font targeted with the relationship with ID `rId10` can be retrieved if needed by using this key and the algorithm above. *end example*]
	
	The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationshi	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
ps	footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

subsetted (Embedded Font Is Subsetted)	Specifies that the embedded font targeted by the id attribute has been subsetted. *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.
	
	If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetted.
	
	[*Example*: Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:
	
	`<w:font w:name="Arial">`
	`…`
	`<w:embedRegular r:id="rId10" w:subsetted="true" />`
	`</w:font>`
	
	The subsetted attribute has a value of `true`, therefore the embedded Arial font targeted with the relationship with ID `rId10` must be treated as a subsetted font. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontRel) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.8.3.7 embedSystemFonts (Embed Common System Fonts)

This element specifies that applications shall embed common system fonts when they are in use and font embedding is enabled for this document using the embedTrueTypeFonts element (§17.8.3.8). *Common system fonts* refer to a set of fonts which are typically always present on a machine, and are not defined by ISO/IEC 29500.

If this element is omitted, then the set of fonts defined as common system fonts should not be embedded in the current document when font embedded is turned on. If the embedTrueTypeFonts element is omitted or `false`, then this setting has no effect.

[*Example*: Consider a WordprocessingML document that specifies that it must embed fonts, including common system fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true" />
<w:embedSystemFonts w:val="true"/>
```

The embedSystemFonts element's val attribute has a value of `true` specifying that common system fonts should be included in this document when they are used. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.8.3.8 embedTrueTypeFonts (Embed TrueType Fonts)

This element specifies that applications shall embed the fonts in use in this document when it is saved. These fonts shall be embedded subject to the algorithm specified in §17.8.1.

If this element is omitted, then fonts in use should not be embedded in the current document.

[*Example*: Consider a WordprocessingML document that specifies that it shall embed fonts, including common system fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true" />
<w:embedSystemFonts w:val="true"/>
```

The embedTrueType element's val attribute has a value of `true` specifying that fonts should be embedded in this document when they are used. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.8.3.9 family (Font Family)

This element specifies the font family of the current font. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If this element is omitted, then its value shall be assumed to be auto.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Calibri">
  <w:family w:val="swiss" />
  …
</w:font>
```

The family element specifies via its val attribute value of swiss that this font is part of the Swiss family. *end example*]

Attributes	Description

val (Font Family Value)	Specifies the font family for the parent font.
	
	[*Example*: Consider the following information stored for a single font:
	
	`<w:font w:name="Times New Roman">`
	`  <w:family w:val="roman" />`
	`  …`
	`</w:font>`
	
	The val attribute value of swiss that this font is part of the Roman family. *end example*]
	
	The possible values for this attribute are defined by the ST_FontFamily simple type (§17.18.30).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontFamily) is located in §A.1. *end note*]

17.8.3.10 font (Properties for a Single Font)

This element specifies the properties for one of the fonts used in this document. A font element shall be written out for each font face used in the document, and includes:

- The name of the font as used in the document's stories
- (optionally) Font metrics allowing other applications to locate appropriate substitute fonts as needed
- (optionally) Embedded forms of the font

[*Example*: Consider the following information stored for a single font:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  <w:charset w:val="00" />
  <w:family w:val="roman" />
  <w:pitch w:val="variable" />
  <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008" w:usb3="00000000"
w:csb0="000001FF" w:csb1="00000000" />
</w:font>
```

The font element contains information about the `Times New Roman` font; specifically, information used to locate a substitute font if it is not available. *end example*]

Attributes	Description

name (Primary Font Name)	Specifies the primary name of the current font. This name shall be used to link the information stored in this element with uses of this value in the rFonts element (§17.3.2.26) in document content.
	
	[*Example*: Consider the following information stored for a single font:
	
	`<w:font w:name="Times New Roman">`
	`  …`
	`</w:font>`
	
	The name attribute specifies that the information contained in this element must be used to look up information about all uses of the `Times New Roman` font in the document contents. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Font) is located in §A.1. *end note*]

17.8.3.11 fonts (Font Table Root Element)

This element specifies the root element for a font table part within a WordprocessingML document, and specifies information about the fonts used in this document, each contained within a child font element.

[*Example*: Consider the following information stored in a font table part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:fonts>
  <w:font w:name="Times New Roman">
    …
  </w:font>
  <w:font w:name="Arial">
    …
  </w:font>
</w:fonts>
```

The fonts element contains information about all fonts used in the document - in this example, the `Times New Roman` and `Arial` fonts. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontsList) is located in §A.1. *end note*]

17.8.3.12 notTrueType (Not a TrueType outline Font)

This element specifies that this font is not a font including TrueType outline in a format conforming to ISO/IEC 14496-22:2007. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then the font shall be assumed to be a font including TrueType outline in a format conforming to ISO/IEC 14496-22:2007.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="JonsFont">
  <w:notTrueType w:val="true" />
  …
</w:font>
```

The notTrueType element specifies via its val attribute value of `true` that this font is not a font including TrueType outline. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.8.3.13 panose1 (Panose-1 Typeface Classification Number)

This element specifies the Panose-1 classification number shown in §5.2.7.17 of ISO/IEC 14496-22. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then no Panose-1 information is available.

[*Example*: Consider the following information stored for a single font:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  …
</w:font>
```

The panose1 element specifies its Panose-1 number via its val attribute value of `02020603050405020304`. *end example*]

Attributes	Description
val (Value)	Specifies the Panose-1 classification number for the font, stored as a series of two digit hexadecimal encodings of each digits of the Panose number. [*Example*: Consider the following information stored for a single font: `<w:panose1 w:val="020F0603050405020304" />` The val attribute specifies that the digits in the Panose-1 number are: 2,15,6,3,5,2,3,4. *end example*] The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Panose) is located in §A.1. *end note*]

17.8.3.14 pitch (Font Pitch)

This element specifies the font pitch of the current font. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then its value shall be assumed to be `default`.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Courier New">
  <w:pitch w:val="fixed" />
  …
</w:font>
```

The pitch element specifies via its val attribute value of `fixed` that this is a fixed width font. *end example*]

Attributes	Description
val (Value)	Specifies the font pitch for the font. [*Example*: Consider the following information stored for a single font:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:pitch w:val="variable" /> The val attribute value of variable specifies that this is a variable width font. *end example*] The possible values for this attribute are defined by the ST_Pitch simple type (§17.18.66).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pitch) is located in §A.1. *end note*]

17.8.3.15 saveSubsetFonts (Subset Fonts When Embedding)

This element specifies that applications shall subset fonts when font embedding is enabled for this document using the embedTrueTypeFonts element (§17.8.3.8). *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.

If this element is omitted, then the set of fonts should not be subsetted in the current document when font embedded is turned on. If the embedTrueTypeFonts element is omitted or false, then this setting has no effect.

[*Example*: Consider a WordprocessingML document that specifies that it must subset embedded fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true"/>
<w:saveSubsetFonts w:val="true"/>
```

The embedSystemFonts element's val attribute has a value of true specifying fonts should be subsetted in this document when they are embedded. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.8.3.16 sig (Supported Unicode Subranges and Code Pages)

This element specifies information identifying the code pages and Unicode subranges for which the parent font provides glyphs using the mechanism defined in §5.2.7.18 and §5.2.7.28 of ISO/IEC 14496-22. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

When storing Unicode subrange information, the appropriate bit in the bitfield shall only be set if the entire subrange is supported by that font.

If this element is omitted, then no supported code page/Unicode subrange information is available.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008" w:usb3="00000000"
w:csb0="000001FF" w:csb1="00000000" />
  …
</w:font>
```

The sig element specifies the supported code pages and Unicode sub ranges via its attributes. For example, the code pages supported are:

- Latin 1
- Latin 2: Eastern Europe
- Cyrillic
- Greek
- Turkish
- Baltic

end example]

Attributes	Description

csb0 (Lower 32 Bits of Code Page Bit Field)	Specifies a four digit hexadecimal encoding of the first 32 bits of the 64-bit code-page bit field that identifies which specific character sets or code pages are supported by the parent font using the format defined by ulCodePageRange1 in §5.2.7.28 of ISO/IEC 14496-22.
	
	[*Example*: Consider font information specified as follows:
	
	`<w:font w:name="Lucida Console">`
	`  <w:sig w:csb0="0000001F" … />`
	`  …`
	`</w:font>`
	
	The csb0 attribute value of `0000001F` specifies that the following code pages are supported by this font:
	• Latin 1
	• Latin 2: Eastern Europe
	• Cyrillic
	• Greek
	• Turkish
	
	end example]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

csb1 (Upper 32 Bits	Specifies a four digit hexadecimal encoding of the upper 32 bits of the 64-bit code-page

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

of Code Page Bit Field)	bit field that identifies which specific character sets or code pages are supported by the parent font using the format defined by ulCodePageRange2 in §5.2.7.28 of ISO/IEC 14496-22.
	
	[*Example*: Consider font information specified as follows:
	
	`<w:font w:name="Lucida Console">`
	`  <w:sig w:csb1="00000000" … />`
	`  …`
	`</w:font>`
	
	The csb1 attribute value of `00000000` specifies that none of the specified code pages are supported by this font. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

usb0 (First 32 Bits of Unicode Subset Bitfield)	Specifies the first 32 bits of the 128-bit Unicode subset bit field (USB) as defined by ulUnicodeRange1 of §5.2.7.18 of ISO/IEC 14496-22.
	
	[*Example*: Consider font information specified as follows:
	
	`<w:font w:name="Times New Roman">`
	`  <w:sig w:usb0="20002A87" … />`
	`  …`
	`</w:font>`
	
	The usb0 attribute value of `20002A87` specifies that the first 32 bits of the bitfield are 00100000000000000010101010000111, which corresponds to:
	• Basic Latin
	• Latin-1 Supplement
	• Latin Extended-A
	• Basic Greek
	• Cyrillic
	• Basic Hebrew
	• Basic Arabic
	• Latin Extended Additional
	
	end example]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

usb1 (Second 32 Bits of Unicode Subset Bitfield)	Specifies the second 32 bits of the 128-bit Unicode subset bit field (USB) as defined by ulUnicodeRange2 of §5.2.7.18 of ISO/IEC 14496-22. [*Example*: Consider font information specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:font w:name="Times New Roman"> <w:sig w:usb1="80000000" … /> … </w:font> The usb0 attribute value of 80000000 specifies that the first 32 bits of the bitfield are 10000000000000000000000000000000, which corresponds to: • Arabic Presentation Forms-A *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
usb2 (Third 32 Bits of Unicode Subset Bitfield)	Specifies the third 32 bits of the 128-bit Unicode subset bit field (USB) as defined by ulUnicodeRange3 of §5.2.7.18 of ISO/IEC 14496-22. [*Example*: Consider font information specified as follows: <w:font w:name="Times New Roman"> <w:sig w:usb2="00000008" … /> … </w:font> The usb0 attribute value of 80000000 specifies that the first 32 bits of the bitfield are 00000000000000000000000000001000, which corresponds to: • Arabic Presentation Forms-B *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).
usb3 (Fourth 32 Bits of Unicode Subset Bitfield)	Specifies the fourth 32 bits of the 128-bit Unicode subset bit field (USB) as defined by ulUnicodeRange4 of §5.2.7.18 of ISO/IEC 14496-22. [*Example*: Consider font information specified as follows: <w:font w:name="Times New Roman"> <w:sig w:usb3="00000000" … /> … </w:font> The usb3 attribute value of 00000000 specifies that the first 32 bits of the bitfield are 00000000000000000000000000000000, which corresponds to no subranges. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontSig) is located in §A.1. *end note*]

17.9 Numbering

Numbering refers to symbols - Arabic numerals, Roman numerals, symbol characters ("bullets"), text strings, etc. - in WordprocessingML that are used to label individual paragraphs of text.

[*Example*: The following two paragraphs each contain numbering as defined by WordprocessingML: the first uses an Arabic numeral, the second a symbol character:

7. This is a paragraph with numbering information.

- This is also a paragraph with numbering information.

end example]

The basis for all numbering in WordprocessingML is specified via two structures:

- abstract numbering definitions
- numbering definition instances

Abstract numbering definitions define the appearance and behavior of a specific set of numbered paragraphs in a document. Because this construct is abstract, they are not be directly referenced by document content, but rather they shall be inherited by a *numbering definition instance*, which itself is referenced by document content.

[*Example*: Consider the following example of an abstract numbering definition in a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:abstractNum w:abstractNumId="4">
  <w:nsid w:val="FFFFFF7F" />
  <w:multiLevelType w:val="singleLevel" />
  <w:lvl w:ilvl="0">
    <w:start w:val="1" />
    <w:lvlText w:val="%1." />
    <w:lvlJc w:val="start" />
    <w:pPr>
      <w:tabs>
        <w:tab w:val="num" w:pos="720" />
      </w:tabs>
      <w:ind w:start="720" w:hanging="360" />
    </w:pPr>
  </w:lvl>
</w:abstractNum>
```

This abstractNum element defines an abstract numbering definition which defines a set of numbering properties. It is inherited by any numbering definition instance which inherits from an abstractNumId equal to 4:

```
<w:num w:numId="2">
    <w:abstractNumId w:val="4" />
</w:num>
```

This num element defines an numbering definition instance which can define overrides to the abstract numbering definition (in this case it does not), and is used by any paragraphs with a numId equal to 2:

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="2" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Level one</w:t>
  </w:r>
</w:p>
```

The resulting paragraph inherits the properties of level 0 in the numbering definition instance of 2 which is simply a instance of the abstract numbering definition of 4. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.9.1 abstractNum (Abstract Numbering Definition)

This element specifies a set of properties which shall dictate the appearance and behavior of a set of numbered paragraphs in a WordprocessingML document. These properties are collectively called an *abstract numbering definition*, and are the basis for all numbering information in a WordprocessingML document.

Although an abstract numbering definition contains a complete set of numbering, it shall not be directly referenced by content (hence the use of *abstract*). Instead, these properties shall be inherited by a numbering definition instance using the num element (§17.9.15), which can then itself be referenced by content.

[*Example*: Consider the following example of an abstractNum in a WordprocessingML document:

```
<w:abstractNum w:abstractNumId="4">
  <w:nsid w:val="FFFFFF7F" />
  <w:multiLevelType w:val="singleLevel" />
  <w:lvl w:ilvl="0">
    <w:start w:val="1" />
    <w:lvlText w:val="%1." />
    <w:lvlJc w:val="start" />
    <w:pPr>
      <w:tabs>
        <w:tab w:val="num" w:pos="720" />
      </w:tabs>
      <w:ind w:start="720" w:hanging="360" />
    </w:pPr>
  </w:lvl>
</w:abstractNum>
```

This abstractNum element defines an abstract numbering definition which must be inherited by any numbering definition instance which inherits from abstract numbering definition with an abstractNumId equal to 4. *end example*]

Attributes	Description
abstractNumId (Abstract Numbering Definition ID)	Specifies a unique number which shall be used as the identifier for this abstract numbering definition. This unique number shall be referenced by any numbering definition instance in order to inherit the properties specified by this abstract numbering definition. [*Example*: Consider the WordprocessingML for an abstract numbering definition with an abstractNumId attribute of 4: <w:abstractNum w:abstractNumId="4"> <w:nsid w:val="FFFFFF7F" /> <w:multiLevelType w:val="singleLevel" /> <w:lvl w:ilvl="0"> <w:start w:val="1" />

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:lvlText w:val="%1." /> <w:lvlJc w:val="start" /> <w:pPr> <w:tabs> <w:tab w:val="num" w:pos="720" /> </w:tabs> <w:ind w:left="720"/> </w:pPr> </w:lvl> </w:abstractNum> The abstractNumId attribute serves as a unique identifier for the abstract numbering definition, allowing numbering definition instances (§17.9.15) with a abstractNumId element with a matching attribute value to inherit the abstract numbering definition properties, for example: <w:numbering> … <w:num w:numId="2"> <w:abstractNumId w:val="0" /> </w:num> <w:num w:numId="3"> <w:abstractNumId w:val="1" /> </w:num> <w:num w:numId="4"> <w:abstractNumId w:val="4" /> </w:num> <w:num w:numId="5"> <w:abstractNumId w:val="4" /> </w:num> </w:numbering> In this case, the final two numbering definition instances both inherit from the abstract numbering definition with a abstractNumId of 4. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_AbstractNum) is located in §A.1. *end note*]

17.9.2 abstractNumId (Abstract Numbering Definition Reference)

This element specifies the abstract numbering definition information whose properties shall be inherited by the parent numbering definition instance.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the WordprocessingML for a document with two numbering definition instances, each referencing a different abstract numbering definition:

```
<w:numbering>
  <w:abstractNum w:abstractNumId="0">
    …
  </w:abstractNum>
  <w:abstractNum w:abstractNumId="1">
    …
  </w:abstractNum>
  …
  <w:num w:numId="1">
    <w:abstractNumId w:val="0" />
  </w:num>
  <w:num w:numId="2">
    <w:abstractNumId w:val="1" />
  </w:num>
  …
</w:numbering>
```

The two numbering definition instances reference the abstract numbering definitions with abstractNumId attribute values of `0` and `1` respectively, via their abstractNumId elements. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.9.3 ilvl (Numbering Level Reference)

This element specifies the numbering level of the numbering definition instance which shall be applied to the parent paragraph.

This numbering level is specified on either the abstract numbering definition's lvl element (§17.9.6), and can be overridden by a numbering definition instance level override's lvl element (§17.9.5).

[*Example*: Consider the following numbered paragraphs in a WordprocessingML document:

1. Level one
    a. Level two

These numbered paragraphs might be represented using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Level one</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="1" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>Level two</w:t>
  </w:r>
</w:p>
```

The WordprocessingML above specifies that the first numbered paragraph references the numbering level of 0, within the numbering definition of the num element (§17.9.15) with a numId attribute equal to 5.

The second numbered paragraph references the numbering of 1, within the same numbering definition instance. The WordprocessingML referenced by the ilvl elements above is given below:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:num w:numId="5">
  <w:abstractNumId w:val="0" />
</w:num>
…
<w:abstractNum w:abstractNumId="0">
  <w:nsid w:val="FFFFFF7F" />
  <w:multiLevelType w:val="singleLevel" />
  <w:lvl w:ilvl="0">
    …
  </w:lvl>
  <w:lvl w:ilvl="1">
    …
  </w:lvl>
</w:abstractNum>
```

In this case, the resulting paragraphs would inherit the properties of the abstract numbering definition levels with ilvl attributes of `0` and `1`, respectively. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.4 isLgl (Display All Levels Using Arabic Numerals)

This element specifies whether or not all levels displayed for a given numbering level's text shall be displayed using the `decimal` number format, regardless of the actual number format of that level in the list. [*Note*: This numbering style is often referred to as the legal numbering style. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is present, then all numbering levels present in the lvlTxt element (§17.9.11) shall be converted to their decimal equivalents when they are displayed in this level in the numbering format. If this element is omitted, then each level is displayed using the numFmt (§17.9.17) of that level.

[*Example*: Consider the numbering set below. In this set of blank numbered paragraphs, three numbering levels have been used and the third has the isLgl property applied, resulting in the following:

```
A
    A.a
    A.b
          1.2.1
          1.2.2
B
    B.a
          2.1.1
```

As shown above, each number in the third level in the list has been converted to its decimal equivalent. The WordprocessingML necessary to turn on the legal numbering rule for the third numbering level is given below:

```
<w:lvl w:ilvl="2">
  …
  <w:isLgl />
  …
</w:lvl>
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.9.5 lvl (Numbering Level Override Definition)

This element specifies the appearance and behavior of a specific numbering level within a given numbering level definition override defined using the lvlOverride element (§17.9.8).

A numbering level override definition is identical to a numbering level definition, except for the fact that it is defined as part of a numbering definition instance using the num element (§17.9.15) rather than as part of an abstract numbering definition using the abstractNum element (§17.9.1).

[*Example*: Consider a numbering definition instance which inherits its information from the abstract numbering definition with abstractNumId of 4, but should use a different set of properties for level 0 of the numbering definition. The resulting WordprocessingML would look like:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:num w:numId="6">
  <w:abstractNumId w:val="4" />
  <w:lvlOverride w:ilvl="0">
    <w:lvl w:ilvl="0">
      <w:start w:val="4" />
      <w:lvlText w:val="%1)" />
      <w:lvlJc w:val="start" />
      <w:pPr>
        <w:ind w:start="360" w:hanging="360" />
      </w:pPr>
    </w:lvl>
  </w:lvlOverride>
</w:num>
```

This numbering definition instance overrides level `0` of the list with the specified numbering level override definition, replacing those in the abstract numbering level definition. *end example*]

[*Note*: The ability to set level overrides optimizes use of numbering in WordprocessingML as it prevents writing out redundant abstract numbering definitions if numbering sets only slightly differ.

Consider using WordprocessingML to create two numbered sets that only differ only in the appearance and style of the first numbering level. Both could use the same abstract numbering definition as long as each references a different numbering definition instance with one of the numbering definition instances leveraging a level override for the first numbering level. Below is WordprocessingML that illustrates this:

```
<w:num w:numId="5">
  <w:abstractNumId w:val="4" />
</w:num>
<w:num w:numId="6">
  <w:abstractNumId w:val="4" />
  <w:lvlOverride w:ilvl="0">
    <w:lvl w:ilvl="0">
      <w:start w:val="4" />
      <w:lvlText w:val="%1)" />
      <w:lvlJc w:val="start" />
      <w:pPr>
        <w:ind w:start="360" w:hanging="360" />
      </w:pPr>
    </w:lvl>
  </w:lvlOverride>
</w:num>
```

end note]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

ilvl (Numbering Level)	Specifies the numbering level definition that is to be defined by this set of numbering properties.
	
	This override is a zero-based index of the number of list levels in the document. [*Example*: A value of 2 is the 3rd list level in the document. *end example*]
	
	[*Example*: Consider the following WordprocessingML for a numbering definition instance:
	
	`<w:num w:numId="6">`
	`  <w:abstractNumId w:val="4" />`
	`  <w:lvlOverride w:ilvl="0">`
	`  …`
	`  </w:lvlOverride>`
	`</w:num>`
	
	In this example, the first numbering level definition (with an ilvl of 0) within the referenced abstract numbering definition is overridden. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

tentative (Tentative Numbering)	Specifies that a given numbering level was been saved by a producer but was not used in the parent document. This means that this numbering level can be redefined by a future consumer without changing the actual content of the document.
	
	A value of 1 or true for this attribute value specifies that the numbering level is not used in the current document's contents.
	
	A value of 0 or false for this attribute value specifies that the numbering level is used in the parent document and cannot be redefined without changing its contents. This is the default value for this attribute, and is implied when this attribute is omitted.
	
	[*Example*: Consider the following WordprocessingML numbering level:
	
	`<w:lvl w:ilvl="0" w:tentative="true" >`
	`  …`
	`</w:lvl>`
	
	This level has the tentative attribute set to true, therefore the contents of this numbering level have not been used in the document and can be redefined by a consumer as desired. *end example*]
	
	If this attribute is equal to 1 or true, the WordprocessingML for a given document contains the numbering level information associated with this numbering level, but the 'tentative' numbering level(s) shall not be represented in any of the hosting application's user interface pertaining to numbering levels.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

tplc (Template Code)	Specifies a unique hexadecimal value which can be used to specify a location within an application's user interface in which this numbering level shall be displayed. The method by which this value is interpreted shall be application-defined.
	
	If this attribute is omitted, then this numbering can be displayed in any location chosen by the consumer.
	
	[*Example*: Consider the following abstract numbering definition:
	
	`<w:abstractNum w:abstractNumId="1" >`
	`…`
	`</w:abstractNum>`
	
	In this example the abstractNum element with attribute abstractNumId equal to 1, would appear in the area within a consumer's application user interface specified by the template code `04090019`. *end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lvl) is located in §A.1. *end note*]

17.9.6 lvl (Numbering Level Definition)

This element specifies the appearance and behavior of a numbering level within a given abstract numbering definition. A numbering level contains a set of properties for the display of the numbering for a given numbering level within an abstract numbering definition.

A numbering level definition is identical to a numbering level override definition, except for the fact that it is defined as part of a numbering definition instance using the abstractNum element (§17.9.1) rather than as part of an abstract numbering definition using the num element (§17.9.15).

[*Example*: Consider the WordprocessingML below:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:abstractNum w:abstractNumId="4">
  <w:nsid w:val="1DE04504" />
  <w:multiLevelType w:val="hybridMultilevel" />
  <w:lvl w:ilvl="0" w:tplc="0409000F">
    …
  </w:lvl>
  <w:lvl w:ilvl="1" w:tplc="04090019">
    …
  </w:lvl>
  <w:lvl w:ilvl="2" w:tplc="04090019">
    …
  </w:lvl>
  <w:lvl w:ilvl="3" w:tplc="0409000F">
    …
  </w:lvl>
  …
</w:abstractNum>
```

This example shows that any paragraph whose numbering properties use the ilvl elements with the attribute val set equal to 0, 1, 2, or 3 has the appearance and behavior of their first four numbered levels specified by the lvl elements given above (assuming that no level overrides have been specified). *end example*]

Attributes	Description

ilvl (Numbering Level)	Specifies the numbering level definition that is to be defined by this set of numbering properties.
	
	This override is a zero-based index of the number of list levels in the document. [*Example*: A value of 2 is the 3rd list level in the document. *end example*]
	
	[*Example*: Consider the following WordprocessingML for a numbering definition instance:
	
	`<w:num w:numId="6">`
	`  <w:abstractNumId w:val="4" />`
	`  <w:lvlOverride w:ilvl="0">`
	`  …`
	`  </w:lvlOverride>`
	`</w:num>`
	
	In this example, the first numbering level definition (with an ilvl of 0) within the referenced abstract numbering definition is overridden. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

tentative (Tentative Numbering)	Specifies that a given numbering level was been saved by a producer but was not used in the parent document. This means that this numbering level can be redefined by a future

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	consumer without changing the actual content of the document.
	
	A value of `1` or `true` for this attribute value specifies that the numbering level is not used in the current document's contents.
	
	A value of `0` or `false` for this attribute value specifies that the numbering level is used in the parent document and cannot be redefined without changing its contents. This is the default value for this attribute, and is implied when this attribute is omitted.
	
	[*Example*: Consider the following WordprocessingML numbering level:
	
	`<w:lvl w:ilvl="0" w:tentative="true" >`
	`…`
	`</w:lvl>`
	
	This level has the tentative attribute set to `true`, therefore the contents of this numbering level have not been used in the document and can be redefined by a consumer as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

tplc (Template Code)	Specifies a unique hexadecimal value which can be used to specify a location within an application's user interface in which this numbering level shall be displayed. The method by which this value is interpreted shall be application-defined.
	
	If this attribute is omitted, then this numbering can be displayed in any location chosen by the consumer.
	
	[*Example*: Consider the following abstract numbering definition:
	
	`<w:abstractNum w:abstractNumId="1" >`
	`…`
	`</w:abstractNum>`
	
	In this example the abstractNum element with attribute abstractNumId equal to `1`, would appear in the area within a consumer's application user interface specified by the template code `04090019`.*end example*]
	
	The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lvl) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.9.7 lvlJc (Justification)

This element specifies the type of justification used on a numbering level's text within a given numbering level. This justification is applied relative to the text margin of the parent numbered paragraph in the document.

If omitted, the paragraph shall have left justification relative to the text margin in left-to-right paragraphs, and right justification relative to the text margin in right-to-left paragraphs.

[*Example*: Consider the numbering level defined below:

```
<w:lvl w:ilvl="8" w:tplc="756C1446" w:tentative="1">
  <w:start w:val="1" />
  <w:numFmt w:val="bullet" />
  <w:lvlText w:val="•" />
  <w:lvlJc w:val="start" />
  …
</w:lvl>
```

In this numbering level, the given numbering symbol is `left` justified with respect to the text margin, therefore the numbering extends left from the text margin towards the text (assuming a left-to-right paragraph). *end example*]

A numbering level's text is the numeral, symbol, character, graphic, etc. used to create a numbered paragraph as defined by the lvlText element (§17.9.11).

[*Example:* Consider the numbered paragraphs below:

1) Example one

a. Example two

- Example three

The numbering symbol in these three numbered paragraphs are "1", "a", and "•", respectively. *end example*]

Attributes	Description
val (Alignment Type)	Specifies the justification which should be applied to the parent object within a document. The possible values (see below) for this attribute are always specified with `left` specifying justification relative to the leading edge of the paragraph, and therefore change semantic between right-to-left and left-to-right documents. [*Example*: Consider the following WordprocessingML fragment for a paragraph in a document: `<w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:jc w:val="end" /> </w:pPr> This paragraph is now right justified on the page for a left-to-right paragraph, left justified for a right-to-left paragraph. *end example*] The possible values for this attribute are defined by the ST_Jc simple type (§17.18.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Jc) is located in §A.1. *end note*]

17.9.8 lvlOverride (Numbering Level Definition Override)

This element specifies an optional override which shall be applied in place of zero or more levels from the abstract numbering definition for a given numbering definition instance. Each instance of this element is used to override the appearance and behavior of a given numbering level definition within the given abstract numbering definition.

[*Example*: Consider a numbering definition instance which inherits its information from the abstract numbering definition with abstractNumId of 4, but wishes to use a different set of properties for level 0 and level 1 of the numbering definition. The resulting WordprocessingML would look like:

```
<w:num w:numId="6">
  <w:abstractNumId w:val="4" />
  <w:lvlOverride w:ilvl="0">
    <w:lvl w:ilvl="0">
      <w:start w:val="4" />
      <w:lvlText w:val="%1)" />
      <w:lvlJc w:val="start" />
      <w:pPr>
        <w:ind w:start="360" w:hanging="360" />
      </w:pPr>
    </w:lvl>
  </w:lvlOverride>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <w:lvlOverride w:ilvl="1">
      <w:lvl w:ilvl="1">
        <w:start w:val="5" />
        <w:lvlText w:val="%Test)" />
        <w:lvlJc w:val="start" />
        <w:pPr>
          <w:ind w:start="360" w:hanging="360" />
        </w:pPr>
      </w:lvl>
    </w:lvlOverride>
  </w:num>
```

end example]

[*Note*: The ability to set level overrides optimizes use of numbering in WordprocessingML as it prevents writing out redundant abstract numbering definitions if numbering sets only slightly differ.

Consider using WordprocessingML to create two numbered sets that only differ only in the appearance and style of the first numbering level. Both could use the same abstract numbering definition as long as each references a different numbering definition instance with one of the numbering definition instances leveraging a level override for the first numbering level. Below is WordprocessingML that illustrates this:

```
  <w:num w:numId="5">
    <w:abstractNumId w:val="4" />
  </w:num>
  <w:num w:numId="6">
    <w:abstractNumId w:val="4" />
    <w:lvlOverride w:ilvl="0">
      <w:lvl w:ilvl="0">
        <w:start w:val="4" />
        <w:lvlText w:val="%1)" />
        <w:lvlJc w:val="start" />
        <w:pPr>
          <w:ind w:start="360" w:hanging="360" />
        </w:pPr>
      </w:lvl>
    </w:lvlOverride>
  </w:num>
```

end note]

Attributes	Description
ilvl (Numbering Level ID)	Specifies the numbering level of a given abstract numbering definition to be overridden. If this number conflicts with the ilvl of the child lvl element, then the latter shall be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	ignored.
	
	[*Example*: Consider a numbering definition instance which inherits its information from the abstract numbering definition with abstractNumId of 4, but wishes to use a different set of properties for level 0 of the numbering definition. The resulting WordprocessingML would look like:
	
	`<w:num w:numId="6">`
	`  <w:abstractNumId w:val="4" />`
	`  <w:lvlOverride w:ilvl="0">`
	`  <w:lvl w:ilvl="0">`
	`  <w:start w:val="4" />`
	`  <w:lvlText w:val="%1)" />`
	`  <w:lvlJc w:val="start" />`
	`  <w:pPr>`
	`  <w:ind w:left="360" />`
	`  </w:pPr>`
	`  </w:lvl>`
	`  </w:lvlOverride>`
	`</w:num>`
	
	This level overrides level 0 of the abstract numbering definition's level properties with the specified set of numbering properties, replacing those in the abstract numbering definition. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumLvl) is located in §A.1. *end note*]

17.9.9 lvlPicBulletId (Picture Numbering Symbol Definition Reference)

This element specifies a picture which shall be used as a numbering symbol for a given numbering level by referring to a picture numbering symbol definition's numPictBullet element (§17.9.20). This reference is made through this element's val attribute.

The picture shall be added to the numbering level by replacing each character in the lvlText with one instance of this image.

[*Example*: Consider the WordprocessingML below illustrating how the lvlPicBulletId references a picture numbering symbol definition though its val attribute:

```
<w:numPicBullet w:numPicBulletId="1">
  <w:drawing>
    …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    </w:drawing>
  </w:numPicBullet>
  …
  <w:abstractNum w:abstractNumId="7">
    <w:nsid w:val="71A06359" />
    <w:multiLevelType w:val="hybridMultilevel" />
    <w:tmpl w:val="10643FE6" />
    <w:lvl w:ilvl="0" w:tplc="B7663E56">
      <w:start w:val="1" />
      <w:numFmt w:val="bullet" />
      <w:lvlText w:val="AA" />
      <w:lvlPicBulletId w:val="1" />
    </w:lvl>
  </w:abstractNum>
```

The resulting numbering must consist of two instances of the image specified using the numPicBullet element. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.10 lvlRestart (Restart Numbering Level Symbol)

This element specifies a one-based index which determines when a numbering level should restart to its start value (§17.9.25). A numbering level restarts when an instance of the specified numbering level, which shall be higher (earlier than this level) or any earlier level is used in the given document's contents. [*Example:* If this value is 2, then both level two and level one reset this value. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, the numbering level shall restart each time the previous numbering level or any earlier level is used. If the specified level is higher than the current level, then this element shall be ignored. As well, a value of `0` shall specify that this level shall never restart.

[*Example*: Consider a set of numbered paragraphs in a WordprocessingML document where numbering level with ilvl of 2 is set to never restart:

```
<w:lvl w:ilvl="0">
  <w:start w:val="1" />
  <w:lvlText w:val="%1)" />
  <w:lvlJc w:val="start" />
   <w:pPr>
    <w:ind w:start="360" w:hanging="360" />
  </w:pPr>
   <w:rPr>
    <w:rFonts w:hint="default" />
  </w:rPr>
</w:lvl>
 <w:lvl w:ilvl="1">
  <w:start w:val="1" />
  <w:numFmt w:val="upperLetter" />
  <w:lvlText w:val="%2)" />
  <w:lvlJc w:val="start" />
   <w:pPr>
    <w:ind w:start="720" w:hanging="360" />
  </w:pPr>
   <w:rPr>
    <w:rFonts w:hint="default" />
  </w:rPr>
</w:lvl>
<w:lvl w:ilvl="2">
  <w:start w:val="1" />
  <w:numFmt w:val="lowerRoman" />
  <w:lvlRestart w:val="0">
  <w:lvlText w:val="%3)" />
  <w:lvlJc w:val="start" />
   <w:pPr>
    <w:ind w:start="1080" w:hanging="360" />
  </w:pPr>
   <w:rPr>
    <w:rFonts w:hint="default" />
  </w:rPr>
</w:lvl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Since the lvlRestart element is omitted in numbering level 1 (a,b,…), the numbering level restarts after numbering level 0 (1,2,…) is used. Numbering level two (i, ii, iii …) never restarts as lvlRestart has a val equal to 0. An example of the resulting content would be as follows:

1) Level one

    a) Level two

        i) Level three

        ii) Level three

2) Level one

    a) Level two

        iii) Level three

        iv) Level four

The resulting set of paragraphs has level two restarting to its start value after each level one (after 2), the next level two is again a)), but level three never restarts and continues at iii) even after the use of a level two and one. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.9.11 lvlText (Numbering Level Text)

This element specifies the textual content which shall be displayed when displaying a paragraph with the given numbering level.

All text in this element's val attribute shall be taken as literal text to be repeated in each instance of this numbering level, except for any use of the percent symbol (%) followed by a number, which shall be used to indicate the one-based index of the number to be used at this level. Any number of a level higher than this level shall be ignored.

When the % syntax is used, the number shall be incremented for each subsequent paragraph of that level (sequential or not), until the restart level is seen between two subsequent paragraphs of this level.

[*Example*: Consider the following WordprocessingML for a numbering level:

```
<w:lvl w:ilvl="1">
  …
  <w:lvlText w:val="StringA %2 StringB %1 StringC %3"/>
  …
</w:lvl>
```

This specifies that three strings (`StringA`, `StringB`, `StringC`) must be used as string literals in the numbering for level two (ilvl of 1) along with the numbering symbol used for level one and level zero. Although level two is also referenced here, it is ignored as it is a higher level than the current numbering level.

Therefore, assuming the numbering symbol used by numbering level zero is an Arabic numeral, and the numbering symbol used by numbering level one is a Roman numeral, a set of numbered paragraphs using this WordprocessingML numbering set must be output as:

```
1
  StringA I StringB 1 StringC
  StringA II StringB 1 StringC
  StringA III StringB 1 StringC
2
  StringA I StringB 2 StringC
  StringA II StringB 2 StringC
```

with the %1 and %2 values corresponding to the current numbering symbol value for numbering level zero and one, respectively. *end example*]

Attributes	Description
null (Level Text Is Null Character)	Specifies that a null character shall be used as the numbering symbol for a given numbering level. If the val attribute contains any content, then this attribute shall be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute is omitted, then the null string shall not be used in place of the empty string. [*Note*: A null character is different from an empty string. *end note*]
	
	[*Example*: Consider the WordprocessingML below:
	
	`<w:lvl w:ilvl="1">`
	`…`
	`<w:lvlText w:null="on" />`
	`…`
	`</w:lvl>`
	
	This level text consists of a single null character, and not the empty string, as the null attribute is set. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

val (Level Text)	Specifies the actual text to be used for the numbering level when it is referenced in the document's content.
	
	If this attribute is not specified, then the empty string shall be used as the level's text.
	
	[*Example*: Consider the WordprocessingML below:
	
	`<w:lvl w:ilvl="1">`
	`…`
	`<w:lvlText w:val="test" />`
	`…`
	`</w:lvl>`
	
	Here the val attribute specifies that the literal string `test` is to be surfaced as the text for the given numbering level, regardless of its position. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LevelText) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.9.12 multiLevelType (Abstract Numbering Definition Type)

This element specifies the type of numbering defined by a given abstract numbering type. This information shall only be used by a consumer to determine user interface behaviors for this numbering definition, and shall not be used to limit the behavior of the list (i.e. a list with multiple levels marked as `singleLevel` shall not be prevented from using levels 2 through 9).

If this element is omitted, then the list shall be assumed to be of any numbering type desired by the consumer.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the WordprocessingML below:

```
<w:abstractNum w:abstractNumId="8">
  …
  <w:multiLevelType w:val="singleLevel" />
  …
</w:abstractNum>
```

This abstract numbering definition is specified to be of the `singleLevel` numbering type by the multiLevelType element. *end example*]

Attributes	Description

val (Abstract Numbering Definition Type)	Specifies the specific type of numbering enabled by a given abstract numbering definition.
	
	[*Example*: Consider the WordprocessingML below:
	
	`<w:abstractNum w:abstractNumId="8">`
	`  …`
	`  <w:multiLevelType w:val="multilevel" />`
	`  …`
	`</w:abstractNum>`
	
	This abstract numbering definition is specified to be of the `multilevel` numbering type, which can be used by consumers to place this numbering correctly within a user interface. *end example*]
	
	The possible values for this attribute are defined by the ST_MultiLevelType simple type (§17.18.58).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MultiLevelType) is located in §A.1. *end note*]

17.9.13 name (Abstract Numbering Definition Name)

This element specifies the name of a given abstract numbering definition. This name can be surfaced in order to provide a user friendly alias for a given numbering definition, but shall not influence the behavior of the list - two identical definitions with different name elements shall behave identically.

If this element is omitted, then this abstract numbering definition shall have no name.

[*Example*: Consider the WordprocessingML below:

```
<w:abstractNum w:abstractNumId="4">
  <w:nsid w:val="5C294B5B" />
  <w:multiLevelType w:val="multilevel" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:tmpl w:val="6F8A81B0" />
    <w:name w:val="Example Name" />
    …
  </w:abstractNum>
```

In this example, the given abstract numbering definition is named `Example Name` by use of the name element. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.9.14 nsid (Abstract Numbering Definition Identifier)

This element associates a unique hexadecimal ID to the parent abstract numbering definition. This number shall be identical for two abstract numbering definitions that are based from the same initial numbering definition - if a document is repurposed and the underlying numbering definition is changed, it shall maintain its original nsid.

If this element is omitted, then the list shall have no nsid and one can be added by a producer arbitrarily.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: This element can be used to determine the abstract numbering definition to be applied to a numbered paragraph copied from one document and pasted into another. Consider a case in which a given numbered paragraph associated with a abstract numbering definition with nsid FFFFFF23, is pasted among numbered paragraphs associated with a completely different appearance and an abstract numbering definition with an nsid of FFFFFF23. Here, because of the distinction enabled by the identical nsid values, the hosting application would not have to arbitrarily keep the pasted numbered paragraph associated with its original abstract numbering definition, as it might use the information provided by the abstract numbering definition's identical nsid values to know that those two numbering sets are identical, and merge the paragraphs into the target numbering format. *end note*]

[*Example*: Consider the WordprocessingML for an abstract numbering definition below:

```
<w:abstractNum w:abstractNumId="3">
  <w:nsid w:val="FFFFFF89" />
  <w:multiLevelType w:val="singleLevel" />
  <w:tmpl w:val="D9842532" />
  …
</w:abstractNum>
```

In this example, the given abstract numbering definition is associated with the unique hexadecimal ID FFFFFF89. *end example*]

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following value for an attribute of simple type ST_LongHexNumber: 00BE2C6C. This value is permitted, as it contains four hexadecimal digits, each an encoding of an octet of the actual decimal number value. It can therefore be interpreted as desired in the context of the parent XML element, *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LongHexNumber) is located in §A.1. *end note*]

17.9.15 num (Numbering Definition Instance)

This element specifies a unique instance of numbering information that can be referenced by zero or more paragraphs within the parent WordprocessingML document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This instance requires the referencing of a base abstract numbering definition through the abstractNumId child element (§17.9.2). This element also can be used to specify a set of optional overrides applied to zero or more levels from the abstract numbering definition inherited by this instance second though the optional lvlOverride child elements (§17.9.8).

[*Example*: Consider the WordprocessingML for a document with four numbering definition instances, two of which reference the same underlying abstract numbering definition:

```
<w:numbering>
  …
  <w:num w:numId="2">
    <w:abstractNumId w:val="0" />
  </w:num>
  <w:num w:numId="3">
    <w:abstractNumId w:val="1" />
  </w:num>
  <w:num w:numId="4">
    <w:abstractNumId w:val="4" />
  </w:num>
  <w:num w:numId="5">
    <w:abstractNumId w:val="4" />
  </w:num>
</w:numbering>
```

As shown above, the first two numbering definition instances reference abstractNumId values of 0 and 1 respectively, and the last two both reference the abstract numbering definition with an abstractNumId of 4. *end example*]

[*Example*: Consider a numbering definition instance which inherits its information from the abstract numbering definition with abstractNumId of 4, but wishes to use a different set of properties for level 0 of the numbering definition. The resulting WordprocessingML would look like:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:num w:numId="6">
  <w:abstractNumId w:val="4" />
  <w:lvlOverride w:ilvl="0">
    <w:lvl w:ilvl="0">
      <w:start w:val="4" />
      <w:lvlText w:val="%1)" />
      <w:lvlJc w:val="start" />
      <w:pPr>
        <w:ind w:start="360" w:hanging="360" />
      </w:pPr>
    </w:lvl>
  </w:lvlOverride>
</w:num>
```

The lvlOverride element specifies an override for level 0 of the abstract numbering definition. *end example*]

Attributes	Description

numId (Numbering Definition Instance ID)	Specifies a unique ID which any numbered paragraph which wishes to inherit these numbering properties shall reference using the numPr element (§17.3.1.19).
	
	[*Example*: Consider the WordprocessingML below for an example numbered paragraph:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:numPr>`
	`  <w:ilvl w:val="0" />`
	`  <w:numId w:val="5" />`
	`  </w:numPr>`
	`  </w:pPr>`
	`  …`
	`</w:p>`
	
	This paragraph references a numbering definition instance with a numId attribute of 5:
	
	`<w:num w:numId="5">`
	`  <w:abstractNumId w:val="4" />`
	`</w:num>`
	
	The numbering definition instance with a numId attribute of 5 correlates with the numbered paragraph with the numbering definition instance referent element with a val of 5, so the numbered paragraph inherits its properties. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Num) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.9.16 numbering (Numbering Definitions)

This element specifies the formatting, display, and functionality of numbering - Arabic numerals, Roman numerals, symbol characters ("bullets"), text strings, etc. - in WordprocessingML documents, which are used to label individual paragraphs of text.

[*Example*: The following two paragraphs each contain numbering as defined by WordprocessingML: the first uses an Arabic numeral, the second a symbol character:

8. This is a paragraph with numbering information.

- This is also a paragraph with numbering information.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Numbering) is located in §A.1. *end note*]

17.9.17 numFmt (Numbering Format)

This element specifies the number format that shall be used to display all numbering at this level in the numbering definition. This information is used to replace the level text string %x, where x is a particular one-based level index, with the appropriate value unless the numFmt value is `bullet`, in which case the literal text of the level text string is used. This value shall be calculated by counting the number of paragraphs at this level since the last restart using the numbering system defined in the val attribute.

When a document has a custom number format specified by the format attribute, it shall use the referenced number format. If the referenced number format cannot be resolved as a number format the consumer shall use the number format specified by the value of the val attribute. If the corresponding value of the val attribute is `custom`, the result is implementation-defined.

If this element is omitted, the level shall be assumed to be of level type `decimal`.

[*Example*: Consider the following WordprocessingML fragment for a numbering level in a numbering definition:

```
<w:lvl w:ilvl="2">
  <w:start w:val="1" />
  <w:numFmt w:val="lowerRoman" />
  <w:lvlRestart w:val="0" />
  <w:lvlText w:val="%3)" />
  <w:lvlJc w:val="start" />
  <w:pPr>
    <w:ind w:start="1080" w:hanging="360" />
  </w:pPr>
  <w:rPr>
    <w:rFonts w:hint="default" />
  </w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</w:lvl>
```

A numFmt value of `lowerLetter` indicates that a consumer must use lowercase letters for all numbering of this level: a,b,c… *end example*]

Attributes	Description
format (Custom Defined Number Format)	Specifies a custom number format using the syntax defined by the XSLT format attribute. This format shall be used for all numbering in the parent object. [*Example*: A value of `ア` indicates that a consumer must use Katakana numbering. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
val (Numbering Format Type)	Specifies the number format that shall be used for all numbering in the parent object. [*Example*: A value of `lowerLetter` indicates that a consumer must use lowercase letters for each number in this grouping: a,b,c… *end example*] The possible values for this attribute are defined by the ST_NumberFormat simple type (§17.18.59).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmt) is located in §A.1. *end note*]

17.9.18 numId (Numbering Definition Instance Reference)

This element specifies the numbering definition instance which shall be used for the given parent numbered paragraph in the WordprocessingML document.

A value of `0` for the val attribute shall never be used to point to a numbering definition instance, and shall instead only be used to designate the removal of numbering properties at a particular level in the style hierarchy (typically via direct formatting).

[*Example*: Consider the WordprocessingML below for an example numbered paragraph:

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="5" />
    </w:numPr>
  </w:pPr>
  …
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This paragraph references a numbering definition instance with a numId attribute of 5, as follows:

```
<w:num w:numId="5">
  <w:abstractNumId w:val="4" />
</w:num>
```

The numbering definition instance reference specifies the given numbering definition instance to be applied to the given paragraph, which itself inherits its properties from abstract numbering definition with abstractNumId of 4. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.19 numIdMacAtCleanup (Last Reviewed Abstract Numbering Definition)

This element specifies to a consumer the progress in the last attempt made by the application to remove unused abstract numbering definitions from a given document. If a legacy document is opened by a consumer, it can choose to remove abstract numbering definition which are 'orphaned' (have no associated numbering definition instances). This element is used by those consumers to indicate their progress (if not complete) in reviewing existing abstract numbering definitions. [*Note*: Removing unused abstract numbering definition from a document reduces the file size, but is not required. *end note*]

If omitted, then all abstract numbering definitions shall be considered reviewed.

[*Example*: Consider a document with 32 abstract numbering definitions, with abstractNumId values ranging from 0 to 85. If an application has only reviewed those abstract numbering definitions with abstractNumId values lower than 25 at save time, it would indicate that state as follows:

```
<w:numIdMacAtCleanup w:val="25"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This value specifies that all abstract numbering definitions with an abstractNumId value higher than 25 have not yet been reviewed. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.20 numPicBullet (Picture Numbering Symbol Definition)

This element specifies the appearance and behavior of a specific picture to be used as the numbering symbol within a numbering level definition in a document, and is the basis for all picture numbering symbol information in a WordprocessingML document.

This element is not used directly within abstract numbering definitions but rather is referenced through its numPicBulletId attribute by the lvlPicBulletId element (§17.9.9) used within numbering level definitions.

[*Example*: Consider the WordprocessingML fragment below which illustrates how a numPicBullet definition is referenced by a picture numbering symbol definition reference through its numPicBulletId attribute:

```
<w:numPicBullet w:numPicBulletId="1">
  <w:drawing>
    …
  </w:drawing>
</w:numPicBullet>
…
<w:abstractNum w:abstractNumId="7">
  <w:nsid w:val="71A06359" />
  <w:multiLevelType w:val="hybridMultilevel" />
  <w:tmpl w:val="10643FE6" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:lvl w:ilvl="0" w:tplc="B7663E56">
      <w:start w:val="1" />
      <w:numFmt w:val="bullet" />
      <w:lvlText w:val="" />
      <w:lvlPicBulletId w:val="1" />
    </w:lvl>
  </w:abstractNum>
```

The lvlPicBulletId element references a numPicBullet element, which defines the size and appearance of all picture bullets of this picture bullet type within the document. It is important to note that this picture bullet can be referenced by multiple levels of various numbering definitions. *end example*]

Attributes	Description

numPicBulletId (Picture Numbering Symbol ID)	Specifies a unique ID for this picture bullet definition which shall be used to reference this picture bullet from a numbering level definition.
	
	[*Example*: Consider the WordprocessingML fragment below which illustrates how a numPicBullet definition is referenced by a picture numbering symbol definition reference through its numPicBulletId attribute:
	
	`<w:numPicBullet w:numPicBulletId="1">`
	`  …`
	`</w:numPicBullet>`
	`…`
	`<w:abstractNum w:abstractNumId="7">`
	`  <w:lvl w:ilvl="0" w:tplc="B7663E56">`
	`  …`
	`  <w:lvlPicBulletId w:val="1" />`
	`  </w:lvl>`
	`</w:abstractNum>`
	
	The lvlPicBulletId element references the ID in the numPicBulletId attribute directly. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumPicBullet) is located in §A.1. *end note*]

17.9.21 numStyleLink (Numbering Style Reference)

This element specifies an abstract numbering that does not contain the actual numbering properties for its numbering type, but rather serves as a reference to a numbering style stored in the document, which shall be applied when this abstract numbering definition is referenced, and itself points at the actual underlying abstract numbering definition to be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The numbering style that is to be applied when this abstract numbering definition is referenced is identified by the string contained in numStyleLink's val attribute.

[*Example*: Consider the abstract numbering definition below:

```
<w:abstractNum w:abstractNumId="0">
  <w:nsid w:val="38901FA4" />
  <w:multiLevelType w:val="multilevel" />
  <w:numStyleLink w:val="TestNumberingStyle" />
</w:abstractNum>
```

This abstract numbering definition references the numbering style with a styleId attribute equal to `TestNumberingStyle`, as follows below:

```
<w:style w:type="numbering" w:styleId="TestNumberingStyle">
  …
</w:style>
```

Therefore, this numbering style must be applied whenever the base abstract numbering definition is inherited by a numbered paragraph. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.9.22 pPr (Numbering Level Associated Paragraph Properties)

This element specifies the paragraph properties which shall be applied as part of a given numbering level within the parent numbering definition. These paragraph properties are applied to any numbered paragraph that references the given numbering definition and numbering level.

Paragraph properties specified on the numbered paragraph itself override the paragraph properties specified by pPr elements within a numbering lvl element (§17.9.5, §17.9.6).

[*Example:* Consider the WordprocessingML below which specifies numbering level paragraph properties:

```
<w:abstractNum w:abstractNumId="1">
  …
  <w:lvl w:ilvl="0">
    …
    <w:pPr>
      <w:tabs>
        <w:tab w:val="num" w:pos="720" />
      </w:tabs>
      <w:ind w:start="720" w:hanging="360" />
    </w:pPr>
  </w:lvl>
</w:abstractNum>
```

Each of the paragraph properties specified inside the pPr element are applied to any numbered paragraph which inherits this numbering level definition as part of the numbering properties in the order defined by the style hierarchy. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrGeneral) is located in §A.1. *end note*]

17.9.23 pStyle (Paragraph Style's Associated Numbering Level)

This element specifies the name of a paragraph style which shall automatically this numbering level when applied to the contents of the document. When a paragraph style is defined to include a numbering definition, any numbering level defined by the numPr element (§17.3.1.19) shall be ignored, and instead this element shall specify the numbering level associated with that paragraph style.

If this element references a style which does not exist, or is not a paragraph style, then it can be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the WordprocessingML below which specifies that the paragraph style with styleId `example`, when applied to paragraphs in the document, must also apply the first numbering level of the abstract numbering definition with an abstractNumId equal to `1`, as follows:

```
<w:abstractNum w:abstractNumId="1">
  …
  <w:lvl w:ilvl="0">
    …
    <w:pStyle w:val="example" />
    <w:pPr>
      <w:tabs>
        <w:tab w:val="num" w:pos="720" />
      </w:tabs>
      <w:ind w:start="720" w:hanging="360" />
    </w:pPr>
    …
  </w:lvl>
</w:abstractNum>
```

The style definition for the paragraph style would only include the numId of the numbering definition instance, and not its level:

```
<w:style w:styleId="example" w:type="paragraph">
  …
  <w:pPr>
    <w:numPr>
      <w:numId w:val="0" />
    </w:numPr>
  </w:pPr>
</w:style>
```

end example]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.9.24 rPr (Numbering Symbol Run Properties)

This element specifies the run properties which shall be applied to the numbering level's text specified in the lvlText element (§17.9.11) when it is applied to paragraphs in this document.

These run properties are applied to all numbering level text used by a given abstract numbering definition and numbering level. It should be noted that run properties specified on a numbered paragraph itself, or on text runs within a numbered paragraph, are separate from the run properties specified by rPr elements within a numbering level, as the latter affects only the numbering text itself, not the remainder of runs in the numbered paragraph.

[*Example*: Consider the WordprocessingML below which uses the rPr element to specify that the numbering symbol used within a given numbering level should be bold and of a 16 point font size:

```
<w:lvl w:ilvl="1">
  …
  <w:rPr>
    <w:b />
    <w:sz w:val="32" />
  </w:rPr>
</w:lvl>
```

The resulting paragraph uses its regular paragraph formatting, but the numbering level text itself must be specifically formatted as bold in 16 point font. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The W3C XML Schema definition of this element's content model (CT_RPr) is located in §A.1. Each child element from the above table shall not occur more than once. [*Note*: This restriction is not reflected in the element's content model due to limitations of W3C XML Schema language. *end note*]

17.9.25 start (Starting Value)

This element specifies the starting value for the numbering used by the parent numbering level within a given numbering level definition. This value is used when this level initially starts in a document, as well as whenever it is restarted via the properties set in the lvlRestart element (§17.9.10).

If this element is omitted, then the starting value shall be zero (0).

[*Example*: Consider the following WordprocessingML fragment for an abstract numbering definition:

```
<w:abstractNum w:abstractNumId="1">
  …
  <w:lvl w:ilvl="0">
    <w:start w:val="2" />
    <w:numFmt w:val="upperLetter"/>
    …
  </w:lvl>
</w:abstractNum>
```

In this example, since upper case Western letters (upperLetter) are being used as numbering symbols for this numbering level, the first instance of a numbering paragraph associated with this abstract numbering definition and numbering level would have the numbering symbol B, the second letter in the number format.

Subsequent numbered paragraphs with this abstract numbering definition and at this level would have their numbering symbols incremented from B (the starting value for this numbering level). *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

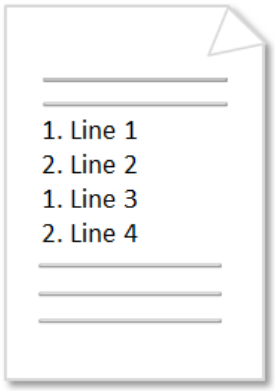
 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.26 startOverride (Numbering Level Starting Value Override)

This element specifies the number that the specified level override shall begin with. This value is used to reset the numbering in a given level.

[*Example*: Consider using WordprocessingML to override the numbering of a particular list item and make the list look as follows:



Firstly, create two w:num instances that reference the same abstract numbering definition. One of the w:num instance overrides the numbering value for the first level. Below is the Wordprocessing ML that illustrates this:

```
<w:num w:numId="5">
  <w:abstractNumId w:val="4"/>
</w:num>

<w:num w:numId="6">
  <w:abstractNumId w:val="4"/>
  <w:lvlOverride w:ilvl="0">
    <w:startOverride w:val = "1"/>
  </w:lvlOverride>
</w:num>
```

Then, in the main document, number the first and the second paragraph as "1" and "2" respectively by using numId 5, and reset the numbering of the third paragraph to "1" by using numId 6. In turn, the numbering of the fourth paragraph becomes "2", because it is incremented based on the new reset value. Below is the WordprocessingML that illustrates this:

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0"/>
      <w:numId w:val="5"/>
    </w:numPr>
  </w:pPr>
  <w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <w:t> Line 1 </w:t>
      </w:r>
    </w:p>
    <w:p>
      <w:pPr>
        <w:numPr>
          <w:ilvl w:val="0"/>
          <w:numId w:val="5"/>
        </w:numPr>
      </w:pPr>
      <w:r>
        <w:t> Line 2 </w:t>
      </w:r>
    </w:p>
    <w:p>
      <w:pPr>
        <w:numPr>
          <w:ilvl w:val="0"/>
          <w:numId w:val="6"/>
        </w:numPr>
      </w:pPr>
      <w:r>
        <w:t> Line 3 </w:t>
      </w:r>
    </w:p>
    <w:p>
      <w:pPr>
        <w:numPr>
          <w:ilvl w:val="0"/>
          <w:numId w:val="5"/>
        </w:numPr>
      </w:pPr>
      <w:r>
        <w:t> Line 4 </w:t>
      </w:r>
    </w:p>
```

end example]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.9.27 styleLink (Numbering Style Definition)

This element specifies that the parent abstract numbering definition is the base numbering definition for the specified numbering style referenced in its val attribute.

If this element is omitted, or it references a style which does not exist, then this numbering definition shall not be the underlying properties for a numbering style.

[*Note*: Numbering styles are never directly referenced by paragraphs or runs in the document – instead, an abstract numbering definition specifies that it contains the underlying numbering information for a numbering style, and one or more numbering definition instances reference a numbering definition which inherits from it. The numbering style itself is just a friendly name on an abstract numbering definition. *end note*]

[E*xample*: Consider the WordprocessingML fragment below, representing an abstract numbering definition which defines the properties for a numbering style:

```
<w:numbering>
  …
  <w:abstractNum w:abstractNumId="5">
    …
    <w:styleLink w:val="ExampleNumberingStyle" />
    …
  </w:abstractNum>
</w:numbering>
…
<w:styles>
  …
  <w:style w:type="numbering" w:styleId="ExampleNumberingStyle">
    <w:name w:val="ExampleNumberingStyle" />
    …
    <w:pPr>
      <w:numPr>
        <w:numId w:val="6" />
      </w:numPr>
    </w:pPr>
  </w:style>
  …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</w:styles>
```

The styleLink element specifies that the abstract numbering definition defines the properties for a numbering style whose styleId matches its val attribute, and is defined in the styles element of the WordprocessingML. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.9.28 suff (Content Between Numbering Symbol and Paragraph Text)

This element specifies the content which shall be added between a given numbering level's text and the text of every numbered paragraph which references that numbering level.

If this element is omitted, then its value shall be assumed to be `tab`.

[*Example:* Consider the numbered paragraph below:

1. Test

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example, a space exists between the numbering symbol `1.` and the numbered paragraph text `Test`. The space would be specified in WordprocessingML as follows:

```
<w:lvl w:ilvl="0">
  …
  <w:suff w:val="space" />
  …
</w:lvl>
```

The suff element with an attribute value of `space` specifies that the character between the numbering's level text and the paragraph text must be a space. *end example*]

Attributes	Description

val (Character Type Between Numbering and Text)	Specifies the character which shall follow the list number.
	
	[*Example:* Consider a numbered for which a tab exists between the numbering symbol and the numbered paragraph's text. The tab would be specified in WordprocessingML as follows:
	
	`<w:lvl w:ilvl="0">`
	`  …`
	`  <w:suff w:val="tab" />`
	`  …`
	`</w:lvl>`
	
	The val attribute with a value of `tab` specifies that the character between the numbering's level text and the paragraph text must be a tab. This tab follows normal tab stop rules. *end example*]
	
	The possible values for this attribute are defined by the ST_LevelSuffix simple type (§17.18.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LevelSuffix) is located in §A.1. *end note*]

17.9.29 tmpl (Numbering Template Code)

This element specifies a unique hexadecimal code which can be used to determine a location within application user interface in which this abstract numbering definition shall be displayed.

If this element is omitted, then this abstract numbering definition can be displayed in any location chosen by the consumer.

[*Example*: Consider the following abstract numbering definition:

```
<w:abstractNum w:abstractNumId="1">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    <w:tmpl w:val="CA48B6BA" />
    …
  </w:abstractNum>
```

In this example the abstractNum element with attribute abstractNumId equal to 1, would appear in the area within a consumer's application user interface specified by the template code CA48B6BA.*end example*]

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following value for an attribute of simple type ST_LongHexNumber: 00BE2C6C. This value is permitted, as it contains four hexadecimal digits, each an encoding of an octet of the actual decimal number value. It can therefore be interpreted as desired in the context of the parent XML element, *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LongHexNumber) is located in §A.1. *end note*]

17.10 Headers and Footers

Headers and footers refer to text, graphics or data (such as page number, date, document title, and so on) that can appear at the top or bottom of each page in a WordprocessingML document.

A header appears in the top margin (above the main document content on the page), while a footer appears in the bottom margin of a document page (below the main document content on the page).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Since WordprocessingML is a flow-based format, headers and footers are applied by specifying the headers and footers for all pages in a particular section of a document.

Within each section of a document there can be up to three different types of headers and footers:

- First page header/footer
- Odd page header/footer
- Even page header/footer

First page headers and footers specify a unique header or footer which shall appear on the first page of a section. Odd page headers and footers specify a unique header and footer which shall appear on all odd numbered pages for a given section. Even page headers and footers specify a unique header and footer which shall appear on all even numbered pages in a given section.

17.10.1 evenAndOddHeaders (Different Even/Odd Page Headers and Footers)

This element specifies whether sections in this document shall have different headers and footers for even and odd pages (an odd page header/footer and an even page header/footer).

If the val attribute is set to `true`, then each section in the document shall use an odd page header for all odd numbered pages in the section, and an even page header for all even numbered pages in the section (counting from the starting value of page numbering for the parent section to determine if the first page is even or odd, as specified with the start attribute on the pgNumType element). If the val attribute is set to `false`, then all pages in a section shall use the odd page header.

This setting does not affect the presence of a first page header on each section, which is specified using the titlePg element (§17.10.6). If a first page header is specified, then all subsequent pages shall have this setting applied, including the first page in the odd/even page count.

If this element is set to `false` and an even page header/footer is specified , then it shall be ignored and only the odd page header/footer shall be displayed. Conversely, if this element is set to true and either header/footer type is omitted for a given section, the appropriate header/footer should be inherited from the previous section; if this is the first section in the document, then a blank header/footer shall be created as needed (another header/footer type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be `false`.

[*Example*: Consider a document which must have a different even and odd page header for each section in its contents. This requirement must be specified using the following WordprocessingML:

```
<w:settings>
  …
  <w:evenAndOddHeaders />
  …
</w:settings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Since the evenAndOddHeaders property is set (and its default value is `true`), this document now has different headers and footers for even and odd pages. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.10.2 footerReference (Footer Reference)

This element specifies a single footer which shall be associated with the current section in the document. This footer shall be referenced via the id attribute, which specifies an explicit relationship to the appropriate Footer part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/footer, is not present, or does not have a TargetMode attribute value of `Internal`, then the document shall be considered non-conformant.

Within each section of a document there can be up to three different types of footers:

- First page footer
- Odd page footer
- Even page footer

The footer type specified by the current footerReference is specified via the type attribute.

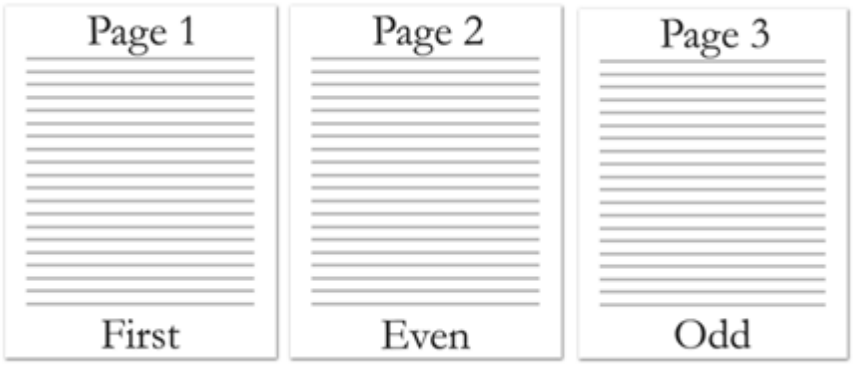
If any type of footer is omitted for a given section, then the following rules shall apply.

- If no footerReference for the first page footer is specified and the titlePg element is specified, then the first page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the titlePg element is not specified, then no first page footer shall be shown, and the odd page footer shall be used in its place.
- If no footerReference for the even page footer is specified and the evenAndOddHeaders element is specified, then the even page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the evenAndOddHeaders element is not specified, then no even page footer shall be shown. and the odd page footer shall be used in its place.
- If no footerReference for the odd page footer is specified then the odd page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created.

[*Example*: Consider a three page document with different first, odd, and even page footers defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



This document defines three footers, each of have a relationship from the document part with a unique relationship ID, as shown in the following packaging markup:

```
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  …
  <Relationship Id="rId6"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/footer" Target="footer1.xml"
/>
  <Relationship Id="rId7"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/footer" Target="footer2.xml"
/>
  <Relationship Id="rId10"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/footer" Target="footer3.xml"
/>
  …
</Relationships>
```

These relationships are then referenced in the section's properties using the following WordprocessingML:

```
<w:sectPr>
  …
  <w:footerReference r:id="rId6" w:type="first" />
  <w:footerReference r:id="rId7" w:type="default" />
  <w:footerReference r:id="rId10" w:type="even" />
  …
</w:sectPr>
```

The resulting section must use the footer part with relationship id `rId6` for the first page, the footer part with relationship id `rId10` for all subsequent even pages, and the footer part with relationship id `rId7` for all subsequent odd pages. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

type (Header or Footer Type)	Specifies the type of header or footer specified by the target relationship ID. This header/footer type determines the page(s) on which the current header or footer shall be displayed.
	
	If any section contains more than a single header or footer of each type, then the document shall be considered non-conformant.
	
	[*Example*: Consider a document with the following WordprocessingML:
	
	`<w:sectPr>`
	`…`
	`<w:footerReference r:id="rId6" w:type="first" />`
	`<w:footerReference r:id="rId7" w:type="first" />`
	`<w:footerReference r:id="rId10" w:type="even" />`
	`…`
	`</w:sectPr>`
	
	The resulting section has two footers of type `first`, and therefore is non-conformant. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider a WordprocessingML section which specifies the following header reference: `<w:headerReference r:id="rId10" w:type="first" />` The resulting section must use the specified header part for the `first` page. *end example*] The possible values for this attribute are defined by the ST_HdrFtr simple type (§17.18.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HdrFtrRef) is located in §A.1. *end note*]

17.10.3 ftr (Footer)

This element specifies the content for a single footer for use within one or more sections of a WordprocessingML document.

Within the ftr element, the content of the element is similar to the content of the body (§17.2.2) element, and contains what is referred to as *block-level markup* - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

[*Example*: Consider the following simple one page document with one footer:

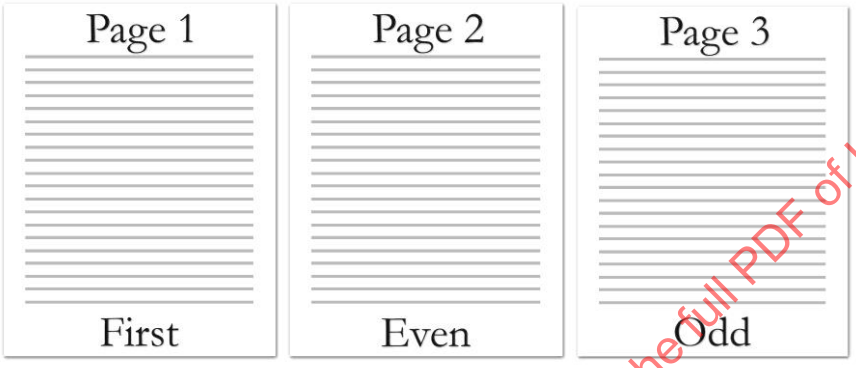
This document defines one footer with the text `footer`. The footer contents are stored in a unique footer part. The resulting footer is represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:ftr>
  <w:p>
    <w:r>
      <w:t>Footer</w:t>
    </w:r>
  </w:p>
</w:ftr>
```

Since footers are containers of block level contents, all block level elements can be used within them. In this particular example, the content is a single paragraph. *end example*]

[*Example*: Consider a more complex three page document with different first, odd, and even page footers defined:



This document defines three footers stored in three different footer parts. The resulting footers are represented by the following WordprocessingML:

First page footer part:

```
<w:ftr>
  <w:p>
    <w:r>
      <w:t>First</w:t>
    </w:r>
  </w:p>
</w:ftr>
```

Even page footer part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:ftr>
  <w:p>
    <w:r>
      <w:t>Even</w:t>
    </w:r>
  </w:p>
</w:ftr>
```

Odd page footer part:

```
<w:ftr>
  <w:p>
    <w:r>
      <w:t>Odd</w:t>
    </w:r>
  </w:p>
</w:ftr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_HdrFtr) is located in §A.1. *end note*]

17.10.4 hdr (Header)

This element specifies the content for a single header for use within one or more sections of a WordprocessingML document.

Within the hdr element, the content of the element is similar to the content of the body (§17.2.2) element, and contains what is referred to as *block-level markup* - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

[*Example*: Consider the following simple one page document with one header:

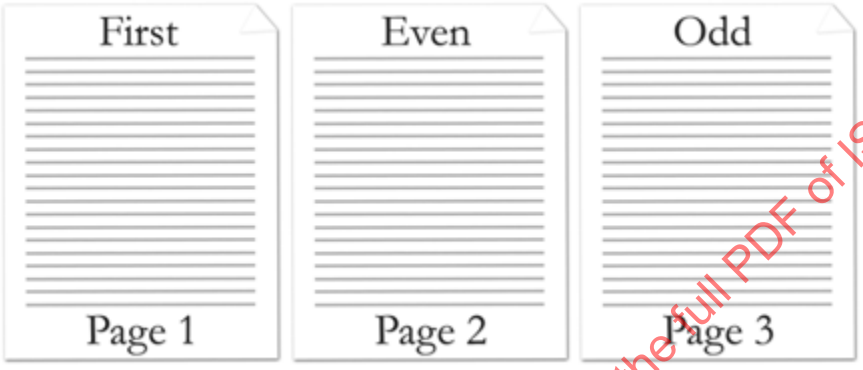
This document defines one header with the text Header. The header's contents is stored in a unique Header part. The resulting header is represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:hdr>
  <w:p>
    <w:r>
      <w:t>Header</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

Since headers are containers of block level contents, all block level elements can be used within them. In this particular example, the content is a single paragraph. *end example*]

[*Example*: Consider a more complex three page document with different first, odd, and even page headers defined:



This document defines three headers stored in three different header parts. The resulting headers are represented by the following WordprocessingML:

First page header part:

```
<w:hdr>
  <w:p>
    <w:r>
      <w:t>First</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

Even page header part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:hdr>
  <w:p>
    <w:r>
      <w:t>Even</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

Odd page header part:

```
<w:hdr>
  <w:p>
    <w:r>
      <w:t>Odd</w:t>
    </w:r>
  </w:p>
</w:hdr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_HdrFtr) is located in §A.1. *end note*]

17.10.5 headerReference (Header Reference)

This element specifies a single header which shall be associated with the current section in the document. This header shall be referenced via the id attribute, which specifies an explicit relationship to the appropriate Header part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/header, is not present, or does not have a TargetMode attribute value of `Internal`, then the document shall be considered non-conformant.

Within each section of a document there can be up to three different types of headers:

- First page header
- Odd page header
- Even page header

The header type specified by the current headerReference is specified via the type attribute.

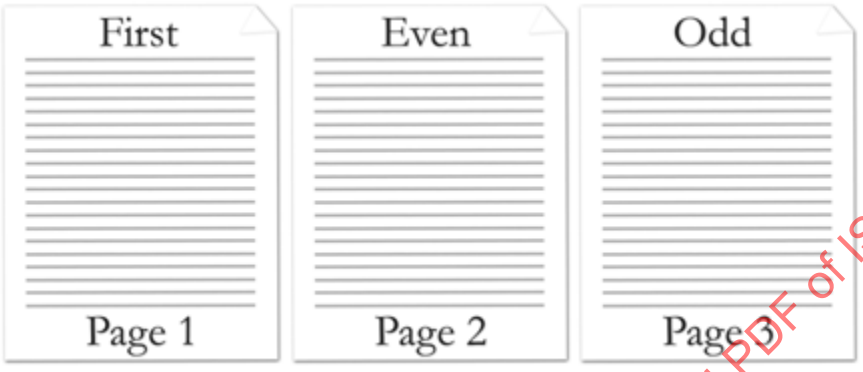
If any type of header is omitted for a given section, then the following rules shall apply.

- If no headerReference for the first page header is specified and the titlePg element is specified, then the first page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the titlePg element is not specified, then no first page header shall be shown, and the odd page header shall be used in its place.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- If no headerReference for the even page header is specified and the evenAndOddHeaders element is specified, then the even page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the evenAndOddHeaders element is not specified, then no even page header shall be shown, and the odd page header shall be used in its place.
- If no headerReference for the odd page header is specified then the odd page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created.

[*Example*: Consider a three page document with different first, odd, and even page header defined as follows:



This document defines three headers, each of have a relationship from the document part with a unique relationship ID, as shown in the following packaging markup:

```
<Relationships
xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  …
  <Relationship Id="rId2"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/header"
Target="header1.xml" />
  <Relationship Id="rId3"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/header"
Target="header2.xml" />
  <Relationship Id="rId5"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/header"
Target="header3.xml" />
  …
</Relationships>
```

These relationships are then referenced in the section's properties using the following WordprocessingML:

```
<w:sectPr>
  …
  <w:headerReference r:id="rId3" w:type="first" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:headerReference r:id="rId5" w:type="default" />
    <w:headerReference r:id="rId2" w:type="even" />
    …
  </w:sectPr>
```

The resulting section must use the header part with relationship id `rId3` for the first page, the header part with relationship id `rId2` for all subsequent even pages, and the header part with relationship id `rId5` for all subsequent odd pages. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
type (Header or Footer Type)	Specifies the type of header or footer specified by the target relationship ID. This header/footer type determines the page(s) on which the current header or footer shall be displayed. If any section contains more than a single header or footer of each type, then the document shall be considered non-conformant. [*Example*: Consider a document with the following WordprocessingML: `<w:sectPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… <w:footerReference r:id="rId6" w:type="first" /> <w:footerReference r:id="rId7" w:type="first" /> <w:footerReference r:id="rId10" w:type="even" /> … </w:sectPr> The resulting section has two footers of type `first`, and therefore is non-conformant. *end example*] [*Example*: Consider a WordprocessingML section which specifies the following header reference: <w:headerReference r:id="rId10" w:type="first" /> The resulting section must use the specified header part for the `first` page. *end example*] The possible values for this attribute are defined by the ST_HdrFtr simple type (§17.18.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HdrFtrRef) is located in §A.1. *end note*]

17.10.6 titlePg (Different First Page Headers and Footers)

This element specifies whether the parent section in this document shall have a different header and footer for its first page.

If the val attribute is set to `true`, then the parent section in the document shall use a first page header for the first page in the section. If the val attribute is set to `false`, then the first page in the parent section shall use the starting value of page numbering for the parent section to determine if the first page should use the even or odd header/footer, as specified with the start attribute on the pgNumType element.

This setting does not affect the presence of even and odd page header on all sections, which is specified using the evenAndOddHeaders element (§17.10.1).

If this element is set to `false` and a first page header/footer is specified , then it shall be ignored and only the odd page header/footer shall be displayed. Conversely, if this element is set to `true` and the first page header/footer type is omitted for the given section, the first page header/footer should be inherited from the previous section; if this is the first section in the document, then a blank header/footer shall be created (another header/footer type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be `false`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a section which must have a different first page header. This requirement is specified using the following WordprocessingML:

```
<w:sectPr>
  …
  <w:titlePg />
  …
</w:sectPr>
```

Since the titlePg property is present (and its default attribute value is `true`), this document now has a different header and footer for its first page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.11 Footnotes and Endnotes

Footnotes and *endnotes* are separate text stories used in documents and books to show the source of borrowed material or to enter explanatory or supplementary information which does not interrupt the normal reading flow of the document.

Footnotes are typically located at the bottom of a page or beneath text being referenced, and *endnotes* are typically placed at the end of a document or at the end of a section. If document has been divided up into one or more sections, each section of a document can contain endnotes.

Both footnotes and endnotes consist of two parts:

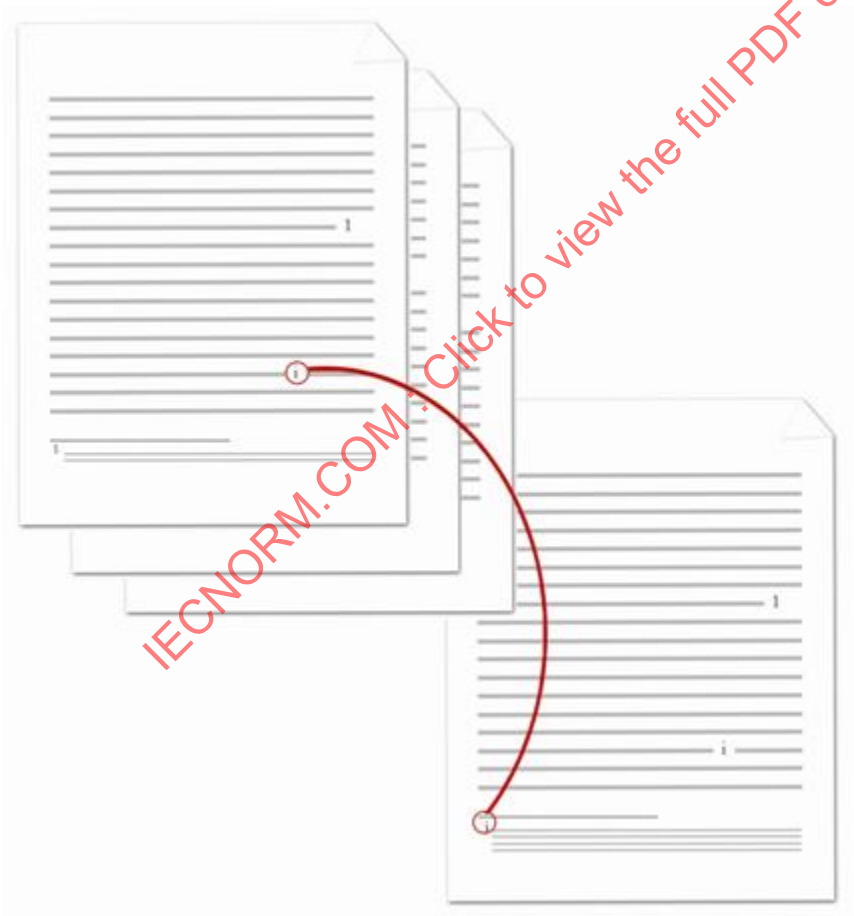
- A note reference mark in the body text to indicate that additional information is in a footnote or endnote, with a numbering system used for each to tell readers whether to look for the note at the end of the page or the end of the document or section.
- The actual footnote or endnote story content.

[*Example:* Example of a footnote applied to text in a document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The note reference mark follows the noted text and specifies that there is associated footnote information, and the footnote itself is at the bottom of the current page. *end example*]

[*Example:* Consider the following example of an endnote applied to text in a document:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

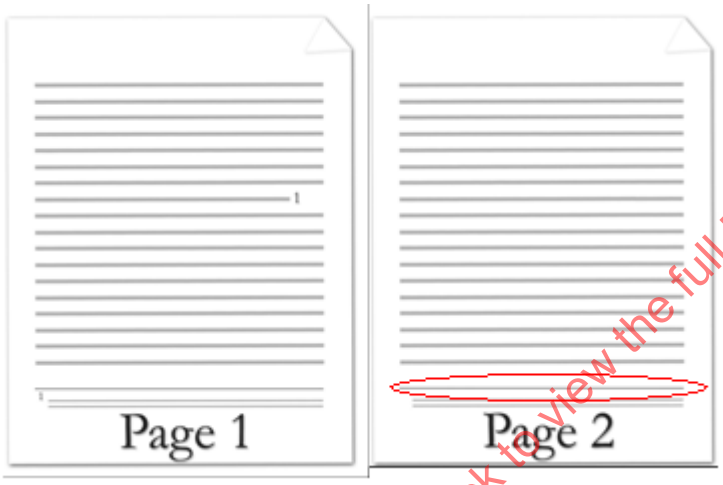
The note reference mark follows the noted text and specifies that there is associated endnote information, and the endnote itself is at the end of the current section. *end example*]

17.11.1 continuationSeparator (Continuation Separator Mark)

This element specifies the presence of a continuation separator mark within the current run. A continuation separator mark is a horizontal line which spans the width of the main story's text extents.

[*Note*: The continuation separator mark is typically used within the context of continuation separator footnotes or endnotes. These footnote and endnote types define the footnote/endnote used to separate the contents of the main document story from continuation of footnotes or endnotes which began on a previous page. *end note*]

[*Example*: Consider the following two pages in a document, where some text is referenced by a footnote that extends to the next page (with the continuation separator circled in red):



The line separating the document text from the footnote that is continued on the next page is represented by the following WordprocessingML:

```
<w:footnote w:type="continuationSeparator" w:id="1">
  <w:p>
    <w:r>
      <w:continuationSeparator />
    </w:r>
  </w:p>
</w:footnote>
```

In this example, the footnote has a content which consists of a single continuationSeparator, which is displayed as a horizontal line across the text extents. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.11.2 endnote (Endnote Content)

This element specifies the content of a single endnote within a WordprocessingML document. Each endnote shall be represented by a single endnote element, which can contain any *block-level content*.

[*Example*: Consider a document with a single endnote, identified by a endnote element, defined in the endnotes part:

```
<w:endnotes>
  <w:endnote w:id="2">
    <w:p>
      <w:pPr>
        <w:pStyle w:val="EndnoteText" />
      </w:pPr>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="EndnoteReference" />
        </w:rPr>
        <w:endnoteRef />
      </w:r>
      <w:r>
        <w:t xml:space="preserve">This is an endnote</w:t>
      </w:r>
    </w:p>
  </w:endnote>
</w:endnotes>
```

This endnote contains an endnote reference mark, as well as the endnote text `This is an endnote`. *end example*]

Attributes	Description

id (Footnote/Endnote ID)	Specifies a unique ID which shall be used to match the contents of a footnote or endnote to the associated footnote/endnote reference mark in the document using the footnoteRef or endnoteRef element, as appropriate.
	
	If this attribute is omitted, then this footnote or endnote shall have no ID. If more than one footnote shares the same ID, then this document shall be considered non-conformant. If more than one endnote shares the same ID, then this document shall be considered non-conformant.
	
	[*Example*: Consider the following footnote as defined in the footnotes part:
	
	`<w:footnotes>`
	`  <w:footnote w:type="normal" w:id="0">`
	`  …`
	`  </w:footnote>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	… </w:footnotes> The contents of this footnote are associated with the footnoteReference with a matching ID, as follows: <w:p> <w:r> <w:footnoteReference w:id="0" /> </w:r> </w:p> The resulting paragraph has a footnote reference mark which references the footnote number value of the footnote with an id of 0. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
type (Footnote/Endnote Type)	Specifies the type of footnote or endnote contained within the current footnote or endnote content definition. If this attribute is omitted, then it shall be considered to be of style normal. If a footnote or endnote is not of style normal, then it shall not be referenced by a footnoteReference or endnoteReference element within the main document story. [*Example*: Consider the following example of a footnote defined in a WordprocessingML document as follows: <w:footnote w:type="continuationSeparator" w:id="1"> <w:p> <w:r> <w:continuationSeparator /> </w:r> </w:p> </w:footnote> In this example, the footnote is of style continuationSeparator and must be used by a consumer to separate continued footnotes from the main document contents (see simple type for full details). *end example*] The possible values for this attribute are defined by the ST_FtnEdn simple type (§17.18.33).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdn) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.11.3 endnote (Special Endnote List)

This element specifies the ID for all endnotes which are located in the current document that are not of style `normal`. Each other type of endnote shall be referenced in this list, or it shall not be loaded. If an endnote is not listed beneath this element, and it is required by the document content, then the document shall be considered non-conformant.

[*Example*: Consider a document that has three endnotes represented by the following WordprocessingML:

```
<w:endnotes …>
  <w:endnote w:type="separator" w:id="0">
    …
  </w:endnote>
  <w:endnote w:type="continuationSeparator" w:id="1">
    …
  </w:endnote>
  <w:endnote w:id="2">
    …
  </w:endnote>
</w:endnotes>
```

Each of the endnotes which are not of style `normal` must be specified in the endnotePr element, as follows:

```
<w:endnotePr>
  <w:endnote w:id="0" />
  <w:endnote w:id="1" />
</w:endnotePr>
```

This indicates to the consumer that the endnotes with an id attribute value of `0` and `1` are special endnotes, and should be treated accordingly. *end example*]

Attributes	Description

id (Footnote/Endnote ID)	Specifies a unique ID that shall be used to match the contents of a footnote or endnote to the associated footnote/endnote reference mark in the document using the footnoteReference or endnoteReference element, as appropriate.
	
	If more than one footnote shares the same ID, then this document shall be considered non-conformant. If more than one endnote shares the same ID, then this document shall be considered non-conformant.
	
	[*Example*: Consider the following footnote as defined in the footnotes part:
	
	`<w:footnotes>`
	`  <w:footnote w:type="normal" w:id="0">`
	`    …`
	`  </w:footnote>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	… </w:footnotes> The contents of this footnote are associated with the footnoteReference with a matching ID, as follows: <w:p> <w:r> <w:footnoteReference w:id="0" /> </w:r> </w:p> The resulting paragraph has a footnote reference mark which references the footnote number value of the footnote with an id of 0. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdnSepRef) is located in §A.1. *end note*]

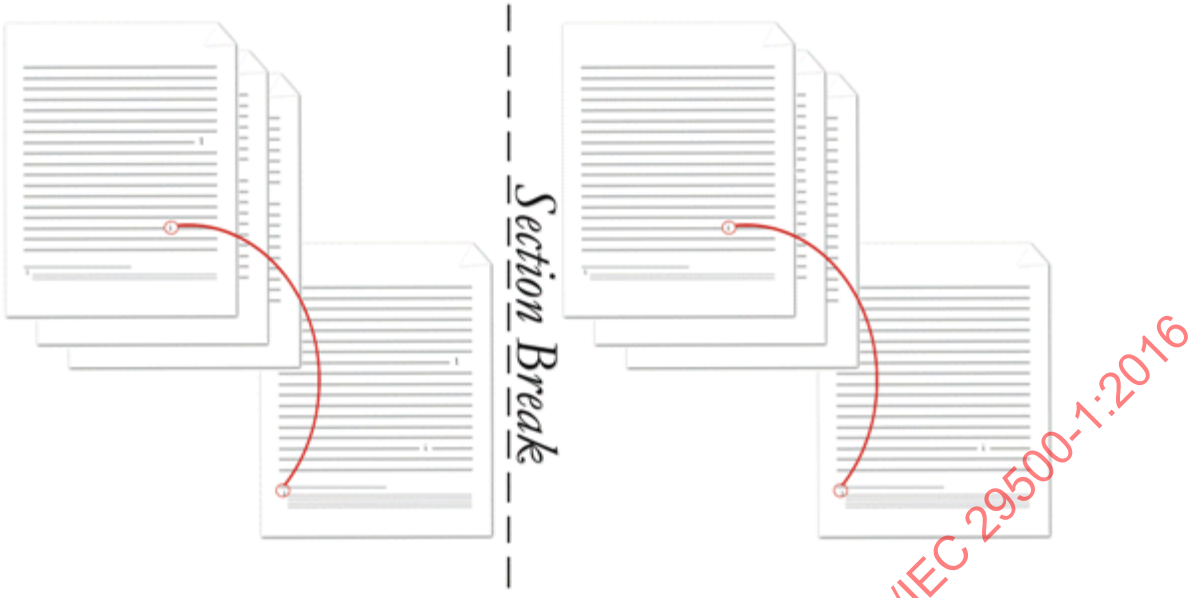
17.11.4 endnotePr (Document-Wide Endnote Properties)

This element specifies the endnote properties for the current document. Each of these properties are stored as a child element within the endnotePr element.

These properties can be overridden for a specific section via the section-wide endnotePr element (§17.11.5).

[*Example*: Consider the following document with two sections, where the endnotes for each section appears at the end of that section and use lower case roman numerals:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



Since both sections are identical, the endnote properties are specified as document-wide level properties (this is not necessary but is most efficient) as follows:

```
<w:settings>
  …
  <w:endnotePr>
    <w:pos w:val="sectEnd"/>
    <w:numFmt w:val="lowerRoman" />
  </w:endnotePr>
  …
</w:settings>
```

Note that the pos element could have been omitted since it is using its default value. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EdnDocProps) is located in §A.1. *end note*]

17.11.5 endnotePr (Section-Wide Endnote Properties)

This element specifies the endnote properties for the current section. Each of these properties are an override of the document-wide endnote properties (§17.11.4) and are stored as a child element within the endnotePr element.

If this element is omitted for a given section, then that section shall use the endnote properties defined at the document-wide level.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a document consisting of three sections, which has endnotes in the first section which use lowercase roman numerals, and endnotes in the third section which use the Chicago Manual of Style format. The WordprocessingML for each section would be specified as follows:

```
<w:sectPr>
  <w:endnotePr>
    <w:numFmt w:val="lowerRoman" />
  </w:endnotePr>
</w:sectPr>
…
<w:sectPr>
  …
</w:sectPr>
…
<w:sectPr>
  …
</w:sectPr>
```

This assumes that the document-wide endnote settings are specified to use the Chicago Manual of Style format, as follows:

```
<w:settings>
  <w:endnotePr>
    <w:numFmt w:val="chicago" />
  </w:endnotePr>
</w:settings>
```

The resulting document would override the endnote numbering format for the first section to `lowerRoman`, but would use the `chicago` endnote numbering format for section three (and would also use it for section two if that section had endnotes. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EdnProps) is located in §A.1. *end note*]

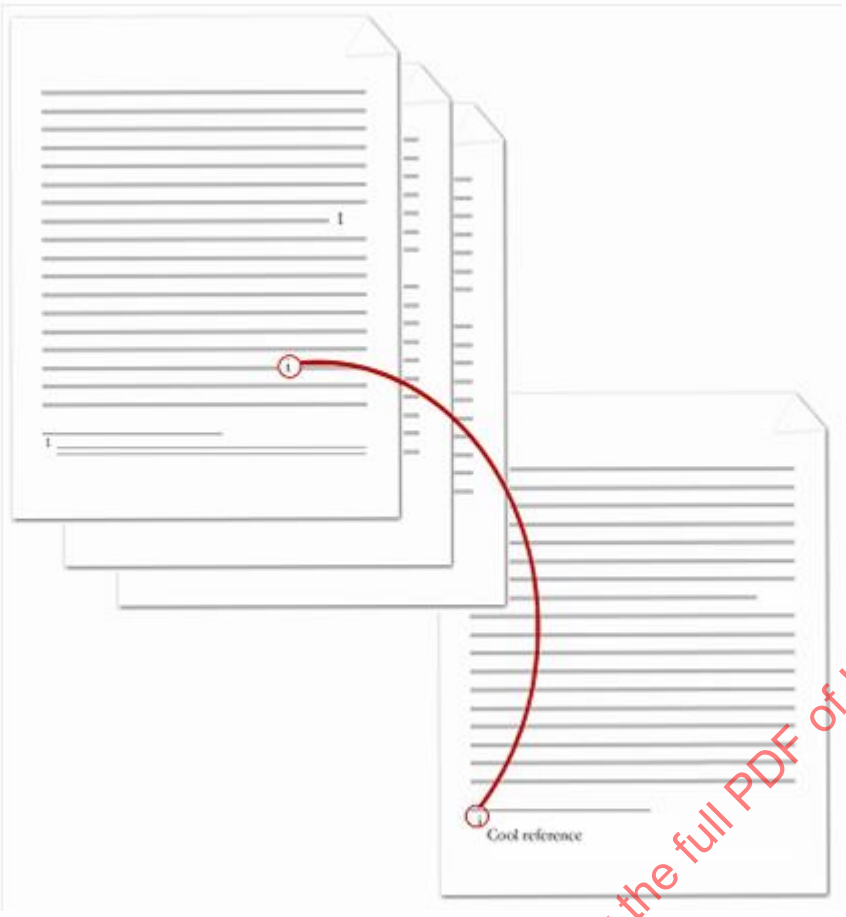
17.11.6 endnoteRef (Endnote Reference Mark)

This element specifies the presence of an endnote reference mark. An *endnote reference mark* is a run of automatically numbered text which follows the numbering format set forth via the numFmt element (§17.11.17).

If an endnote reference mark is specified within a run which is not part of an endnote, then that endnote reference mark can be ignored.

[*Example*: Consider the following document where some text is referenced by an endnote at the end of the document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



The endnote reference mark is the lower case roman numeral within the actual endnote itself in the diagram above. The contents of the endnote (including the endnote reference mark) are represented by the following WordprocessingML:

```
<w:endnote w:id="2">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="EndnoteText" />
    </w:pPr>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="EndnoteReference" />
      </w:rPr>
      <w:endnoteRef />
    </w:r>
    <w:r>
      <w:t>Cool reference</w:t>
    </w:r>
  </w:p>
</w:endnote>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The resulting endnote contains the literal endnote content of `Cool reference`, preceding by an automatically numbered endnote reference mark. Since this is the first endnote in the document, that automatically numbered reference mark uses the lower case roman numeral `i`. *end example*]

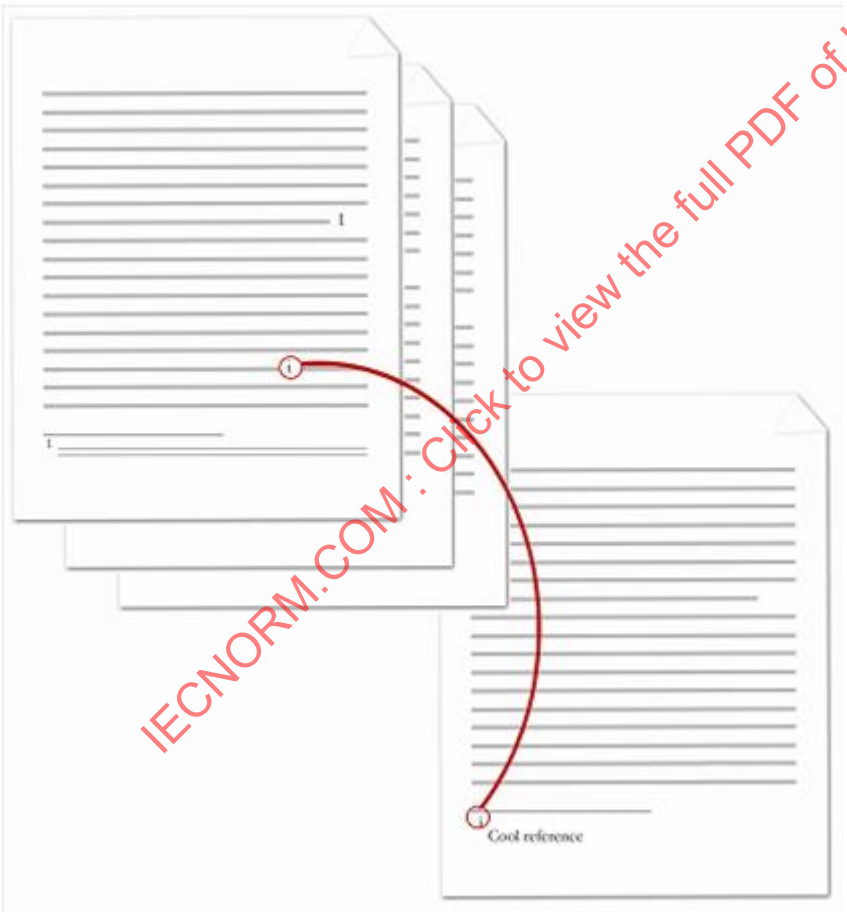
[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.11.7 endnoteReference (Endnote Reference)

This element specifies the presence of an endnote reference. An *endnote reference* is a run of automatically numbered text which references a particular endnote within the parent document, and inherits the endnote reference mark's numbering.

If an endnote reference is specified within a footnote or endnote, then the document shall be considered non-conformant.

[*Example*: Consider the following document where some text is referenced by an endnote at the end of the document:



The endnote reference is the lower case roman numeral within the document content in the diagram above. The contents of the paragraph which contains the endnote reference are represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:p>
  <w:r>
    <w:t>This text is followed by an endnote</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="EndnoteReference" />
    </w:rPr>
    <w:endnoteReference w:id="2" />
  </w:r>
  <w:r>
    <w:t>.</w:t>
  </w:r>
</w:p>
```

The resulting paragraph contains the literal text content of `This text is followed by an endnote`, followed by an automatically numbered endnote reference. Since this is the first endnote in the document, that automatically numbered reference inherits the lower case roman numeral `i` from the endnote reference mark. *end example*]

Attributes	Description

customMarkFollows (Suppress Footnote/Endnote Reference Mark)	Specifies that the current footnote or endnote shall not have an associated footnote or endnote reference mark, as appropriate.
	
	This attribute shall be used to specify that a particular footnote or endnote shall not increment the numbering for its associated footnote/endnote numbering format, so that the use of a footnote with a custom footnote mark does not cause a missing value in the footnote/endnote values. The display of the mark is specified via the footnoteRef/endnoteRef elements, as appropriate.
	
	If this attribute is omitted, then the footnote or endnote reference mark shall not be skipped when incrementing over this footnote or endnote.
	
	[*Example*: Consider a footnote with an `id` value of `1` that uses a custom footnote mark:
	
	`<w:footnotes>`
	`<w:footnote w:id="0">`
	…
	`</w:footnote>`
	`<w:footnote w:id="2">`
	…
	`</w:footnote>`
	`<w:footnote w:id="2">`
	…
	`</w:footnote>`
	`</w:footnotes>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If the numbering format for footnotes in this document is upperRoman, then the first footnote must be I, the second is suppressed, and the third is II, noticing that the second does not increment the numbering sequence. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

id (Footnote/Endnote ID Reference)	Specifies the footnote or endnote which is being referenced by the current footnote or endnote reference in the document.
	
	If the resulting footnote or endnote ID is not present in the footnotes or endnote part (as appropriate), then this document shall be considered non-conformant.
	
	[*Example*: Consider a paragraph with an endnote reference, represented by the following WordprocessingML:
	
	`<w:p>`
	`  <w:r>`
	`  <w:t>This text is followed by an endnote</w:t>`
	`  </w:r>`
	`  <w:r>`
	`  <w:endnoteReference w:id="2" />`
	`  </w:r>`
	`  <w:r>`
	`  <w:t>.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text references the endnote in the document's endnotes part which has an id value of 2. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdnRef) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.11.8 endnotes (Document Endnotes)

This element specifies the set of all endnotes in the document, including endnote separators and continuation notices. This element is the root node for the Endnotes part.

[*Example*: Consider the following example of the contents of the endnotes part:

```
<w:endnotes>
  <w:endnote w:type="separator" w:id="0">
    …
  </w:endnote>
  <w:endnote w:type="continuationSeparator" w:id="1">
    …
  </w:endnote>
  <w:endnote w:id="2">
    …
  </w:endnote>
</w:endnotes>
```

The endnotes part contains the definition for one normal endnote, as well as the separator and continuation separator endnote for this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Endnotes) is located in §A.1. *end note*]

17.11.9 footnote (Special Footnote List)

This element specifies the ID for all footnotes which are located in the current document that are not of style `normal`. Each other type of footnote shall be referenced in this list, or it shall not be loaded. This means that if a special footnote is not listed beneath this element, and it is required by the document content, then the document shall be considered non-conformant.

[*Example*: Consider a document that has three footnotes represented by the following WordprocessingML:

```
<w:footnotes>
  <w:footnote w:type="separator" w:id="0">
    …
  </w:footnote>
  <w:footnote w:type="continuationSeparator" w:id="1">
    …
  </w:footnote>
  <w:footnote w:id="2">
    …
  </w:footnote>
</w:footnotes>
```

Each of the footnotes which are not of style `normal` must be specified in the footnotePr element, as follows:

```
<w:footnotePr>
  <w:footnote w:id="0" />
  <w:footnote w:id="1" />
</w:footnotePr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This indicates to the consumer that the footnotes with an id attribute value of 0 and 1 are special footnotes, and should be treated accordingly. *end example*]

Attributes	Description

id (Footnote/Endnote ID)	Specifies a unique ID which shall be used to match the contents of a footnote or endnote to the associated footnote/endnote reference mark in the document using the footnoteRef or endnoteRef element, as appropriate.
	
	If more than one footnote shares the same ID, then this document shall be considered non-conformant. If more than one endnote shares the same ID, then this document shall be considered non-conformant.
	
	[*Example*: Consider the following footnote as defined in the footnotes part:
	
	`<w:footnotes>`
	`  <w:footnote w:type="normal" w:id="0">`
	`  …`
	`  </w:footnote>`
	`  …`
	`</w:footnotes>`
	
	The contents of this footnote are associated with the footnoteReference with a matching ID, as follows:
	
	`<w:p>`
	`  <w:r>`
	`  <w:footnoteReference w:id="0" />`
	`  </w:r>`
	`</w:p>`
	
	The resulting paragraph has a footnote reference mark which references the footnote number value of the footnote with an id of 0. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdnSepRef) is located in §A.1. *end note*]

17.11.10 footnote (Footnote Content)

This element specifies the content of a single footnote within a WordprocessingML document. Each footnote shall be represented by a single footnote element, which can contain any *block-level content*.

[*Example*: Consider a document with a single footnote, identified by a footnote element, defined in the footnotes part as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:footnotes>
  <w:footnote w:id="2">
    <w:p>
      <w:pPr>
        <w:pStyle w:val="FootnoteText" />
      </w:pPr>
      <w:r>
        <w:rPr>
          <w:rStyle w:val="FootnoteReference" />
        </w:rPr>
        <w:footnoteRef />
      </w:r>
      <w:r>
        <w:t xml:space="preserve">This is a sample footnote</w:t>
      </w:r>
    </w:p>
  </w:footnote>
</w:footnotes>
```

This footnote contains an footnote reference mark, as well as the endnote text `This is a sample footnote`. *end example*]

Attributes	Description

id (Footnote/Endnote ID)	Specifies a unique ID that shall be used to match the contents of a footnote or endnote to the associated footnote/endnote reference mark in the document using the footnoteReference or endnoteReference element, as appropriate.
	
	If this attribute is omitted, then this footnote or endnote shall have no ID. If more than one footnote shares the same ID, then this document shall be considered non-conformant. If more than one endnote shares the same ID, then this document shall be considered non-conformant.
	
	[*Example*: Consider the following footnote as defined in the footnotes part:
	
	`<w:footnotes>`
	`  <w:footnote w:type="normal" w:id="0">`
	`    …`
	`  </w:footnote>`
	`  …`
	`</w:footnotes>`
	
	The contents of this footnote are associated with the footnoteReference with a matching ID, as follows:
	
	`<w:p>`
	`  <w:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:footnoteReference w:id="0" /> </w:r> </w:p> The resulting paragraph has a footnote reference mark which references the footnote number value of the footnote with an id of 0. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
type (Footnote/Endnote Type)	Specifies the type of footnote or endnote contained within the current footnote or endnote content definition. If this attribute is omitted, then it shall be considered to be of style normal. If a footnote or endnote is not of style normal, then it shall not be referenced by a footnoteReference or endnoteReference element within the main document story. [*Example*: Consider the following example of a footnote defined in a WordprocessingML document as follows: <w:footnote w:type="continuationSeparator" w:id="1"> <w:p> <w:r> <w:continuationSeparator /> </w:r> </w:p> </w:footnote> In this example, the footnote is of style continuationSeparator and must be used by a consumer to separate continued footnotes from the main document contents (see simple type for full details). *end example*] The possible values for this attribute are defined by the ST_FtnEdn simple type (§17.18.33).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdn) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.11.11 footnotePr (Section-Wide Footnote Properties)

This element specifies the footnote properties for the current section. Each of these properties are an override of the document-wide footnote properties (§17.11.12) and are stored as a child element within the footnotePr element.

If this element is omitted for a given section, then that section shall use the footnote properties defined at the document-wide level.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a document consisting of three sections, which has footnotes in the first section which appear below text, and footnotes in the third section which appear at the bottom of the page. The WordprocessingML for each section would be specified as follows:

```
<w:sectPr>
  <w:footnotePr>
    <w:pos w:val="beneathText" />
  </w:footnotePr>
</w:sectPr>
…
<w:sectPr>
  …
</w:sectPr>
…
<w:sectPr>
  …
</w:sectPr>
```

This assumes that the document-wide footnote settings are specified as the default positioning at the bottom of the page by omitting the pos element (§17.11.21), as follows:

```
<w:settings>
  <w:footnotePr>
    …
  </w:footnotePr>
</w:settings>
```

The resulting document would override the footnote positioning for the first section to `beneathText`, but would use the `pageBottom` footnote positioning for section three (and would also use it for section two if that section had footnotes. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnProps) is located in §A.1. *end note*]

17.11.12 footnotePr (Document-Wide Footnote Properties)

This element specifies the footnote properties for this document. Each property is stored as a unique element within the footnotePr element.

These properties can be overridden for a specific section via the section-wide footnotePr element (§17.11.11).

[*Example*: Consider the following one page document, where the footnote appears beneath the text it references:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Since the document consists of a single footnote, the footnote properties can be stored in either the section-wide or document-wide footnote properties. Assuming that they are stored in the latter, the footnote properties are represented by the following WordprocessingML:

```
<w:settings>
  …
  <w:footnotePr>
    <w:pos w:val="beneathText" />
  </w:footnotePr>
  …
</w:settings>
```

The footnote properties specify that footnotes appear below the noted text *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnDocProps) is located in §A.1. *end note*]

17.11.13 footnoteRef (Footnote Reference Mark)

This element specifies the presence of a footnote reference mark. A *footnote reference mark* is a run of automatically numbered text which follows the numbering format set forth via the footnote numFmt element (§17.11.18).

If a footnote reference mark is specified within a run which is not part of a footnote, then that footnote reference mark can be ignored.

[*Example*: Consider the following document where some text is referenced by a footnote at the end of the page:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The footnote reference mark is the decimal number within the actual footnote itself in the image above. The contents of the footnote (including the footnote reference mark) are represented by the following WordprocessingML:

```
<w:footnote w:id="2">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="FootnoteText" />
    </w:pPr>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="FootnoteReference" />
      </w:rPr>
      <w:footnoteRef />
    </w:r>
    <w:r>
      <w:t>Cool reference</w:t>
    </w:r>
  </w:p>
</w:footnote>
```

The resulting footnote contains the literal endnote content of `Cool reference`, preceding by an automatically numbered footnote reference mark. Since this is the first footnote in the document, that automatically numbered reference mark uses the first decimal number `1`. It is also important to note that the use of styles `FootnoteText` and `FootnoteReference` is not required, these can simply be added by a particular producer automatically to give the footnote contents are particular style (just like any other use of styles). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.11.14 footnoteReference (Footnote Reference)

This element specifies the presence of a footnote reference. A *footnote reference* is a run of automatically numbered text which references a particular footnote within the parent document, and inherits the footnote reference mark's numbering.

If an footnote reference is specified within a footnote or endnote, then the document shall be considered non-conformant.

[*Example*: Consider the following document where some text is referenced by a footnote at the bottom of the page:

The footnote reference is the superscript decimal number within the document content in the diagram above. The contents of the paragraph which contains the footnote reference are represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:p>
  <w:r>
    <w:t>Some referenced text.</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="FootnoteReference" />
    </w:rPr>
    <w:footnoteReference w:id="2" />
  </w:r>
</w:p>
```

The resulting paragraph contains the literal text content of `Some referenced text.`, followed by an automatically numbered footnote reference. Since this is the first footnote in the document, that automatically numbered reference inherits the decimal number `1` from the footnote reference mark. *end example*]

Attributes	Description

customMarkFollows (Suppress Footnote/Endnote Reference Mark)	Specifies that the current footnote or endnote shall not have an associated footnote or endnote reference mark, as appropriate.
	
	This attribute shall be used to specify that a particular footnote or endnote shall not increment the numbering for its associated footnote/endnote numbering format, so that the use of a footnote with a custom footnote mark does not cause a missing value in the footnote/endnote values. The display of the mark is specified via the footnoteRef/endnoteRef elements, as appropriate.
	
	If this attribute is omitted, then the footnote or endnote reference mark shall not be skipped when incrementing over this footnote or endnote.
	
	[*Example*: Consider a footnote with an `id` value of `1` that uses a custom footnote mark:
	
	`<w:footnotes>`
	`<w:footnote w:id="0">`
	`…`
	`</w:footnote>`
	`<w:footnote w:id="2">`
	`…`
	`</w:footnote>`
	`<w:footnote w:id="2">`
	`…`
	`</w:footnote>`
	`</w:footnotes>`
	
	If the numbering format for footnotes in this document is `upperRoman`, then the first footnote must be `I`, the second is suppressed, and the third is `II`, noticing that the second does not increment the numbering sequence. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

id (Footnote/Endnote ID Reference)	Specifies the footnote or endnote which is being referenced by the current footnote or endnote reference in the document.
	
	If the resulting footnote or endnote ID is not present in the footnotes or endnote part (as appropriate), then this document shall be considered non-conformant.
	
	[*Example*: Consider a paragraph with an endnote reference, represented by the following WordprocessingML:
	
	`<w:p>`
	`  <w:r>`
	`  <w:t>This text is followed by an endnote</w:t>`
	`  </w:r>`
	`  <w:r>`
	`  <w:endnoteReference w:id="2" />`
	`  </w:r>`
	`  <w:r>`
	`  <w:t>.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	This text references the endnote in the document's endnotes part which has an id value of 2. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnEdnRef) is located in §A.1. *end note*]

17.11.15 footnotes (Document Footnotes)

This element specifies the set of all footnotes in the document, including footnote separators and continuation notices. This element is the root node for the Footnotes part.

[*Example*: Consider the following example of the contents of the footnotes part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:footnotes>
  <w:footnote w:type="separator" w:id="0">
    …
  </w:footnote >
  <w:footnote w:type="continuationSeparator" w:id="1">
    …
  </w:footnote>
  <w:footnote w:id="2">
    …
  </w:footnote>
</w:footnotes>
```

The footnotes part contains the definition for one normal footnote, as well as the separator and continuation separator footnotes for this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Footnotes) is located in §A.1. *end note*]

17.11.16 noEndnote (Suppress Endnotes In Document)

This element specifies that all endnotes in this document shall not be displayed or printed. If this element is placed on any section break other than the first section break in the document, it shall be ignored.

If this element is omitted, endnotes shall not be suppressed in the current document.

[*Example*: Consider a document in which in the first section endnotes are marked to be hidden:

```
<w:sectPr>
  <w:noEndnote />
</w:sectPr>
```

In this example, this document does not display endnotes. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.11.17 numFmt (Endnote Numbering Format)

This element specifies the numbering format that shall be used to determine the endnote reference mark value for all automatically numbered endnote reference marks (those without the suppressRef attribute set).

If this element is omitted, then the numbering format shall be assume to be `decimal`.

[*Example*: Consider the following footnote reference with the number format set to upper case letters:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This footnote numbering format is specified by the following WordprocessingML:

```
<w:footnotePr>
  <w:numFmt w:val="upperLetter" />
</w:footnotePr>
```

end example]

Attributes	Description
format (Custom Defined Number Format)	Specifies a custom number format using the syntax defined by the XSLT format attribute. This format shall be used for all numbering in the parent object. [*Example*: A value of `ア` indicates that a consumer must use Katakana numbering. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
val (Numbering Format Type)	Specifies the number format that shall be used for all numbering in the parent object. [*Example*: A value of `lowerLetter` indicates that a consumer must use lowercase letters for each number in this grouping: a,b,c… *end example*] The possible values for this attribute are defined by the ST_NumberFormat simple type (§17.18.59).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmt) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.11.18 numFmt (Footnote Numbering Format)

This element specifies the numbering format that shall be used to determine the footnote reference mark value for all automatically numbered footnote reference marks (those without the suppressRef attribute set).

If this element is omitted, then the numbering format shall be assume to be `decimal`.

[*Example*: Consider the following footnote reference with the number format set to upper case letters:

This footnote numbering format is specified by the following WordprocessingML:

```
<w:footnotePr>
  <w:numFmt w:val="upperLetter" />
</w:footnotePr>
```

end example]

Attributes	Description
format (Custom Defined Number Format)	Specifies a custom number format using the syntax defined by the XSLT format attribute. This format shall be used for all numbering in the parent object. [*Example*: A value of `ア` indicates that a consumer must use Katakana numbering. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
val (Numbering Format Type)	Specifies the number format that shall be used for all numbering in the parent object. [*Example*: A value of `lowerLetter` indicates that a consumer must use lowercase letters for each number in this grouping: a,b,c… *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_NumberFormat simple type (§17.18.59).

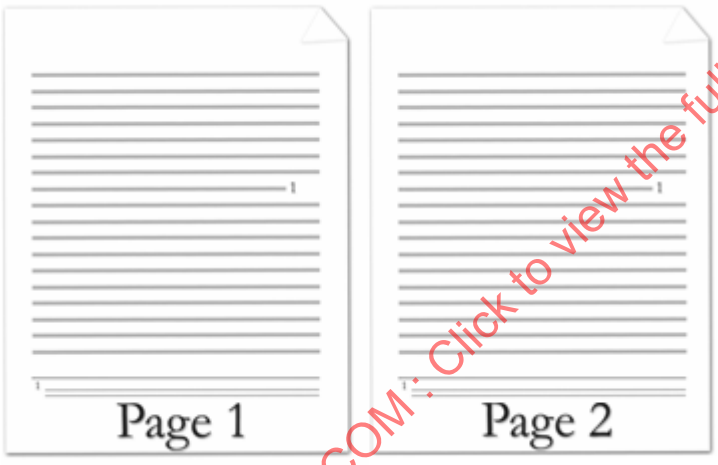
[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmt) is located in §A.1. *end note*]

17.11.19 numRestart (Footnote and Endnote Numbering Restart Location)

This element specifies when all automatic numbering for the footnote or endnote reference marks shall be restarted. When restarted, the next automatically numbered footnote or endnote in the document (each footnote/endnote type is handled independently) shall restart to the specified numStart value (§17.11.20).

If this element is omitted, then automatic numbering shall not be restarted between each page or section (a vlaue of `continuous`).

[*Example*: Consider the following two page document where the numbering must be reset after each page to its starting value:



The footnote automatic restarting of the numbering is represented by the following WordprocessingML:

```
<w:footnotePr>
  <w:numRestart w:val="eachPage" />
</w:footnotePr>
```

end example]

Attributes	Description
val (Automatic Numbering Restart Value)	Specifies when the automatic numbering shall be restarted for the current set of footnotes or endnotes.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider a WordprocessingML document where the numbering for its endnotes must be restarted after each section must be restarted after each page. This setting is represented by the following WordprocessingML:
	
	`<w:footnotePr>`
	`  <w:numRestart w:val="eachSect" />`
	`</w:footnotePr>`
	
	The val attribute value of `eachSect` specifies that numbering shal be restarted after each section. *end example*]
	
	The possible values for this attribute are defined by the ST_RestartNumber simple type (§17.18.74).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumRestart) is located in §A.1. *end note*]

17.11.20 numStart (Footnote and Endnote Numbering Starting Value)

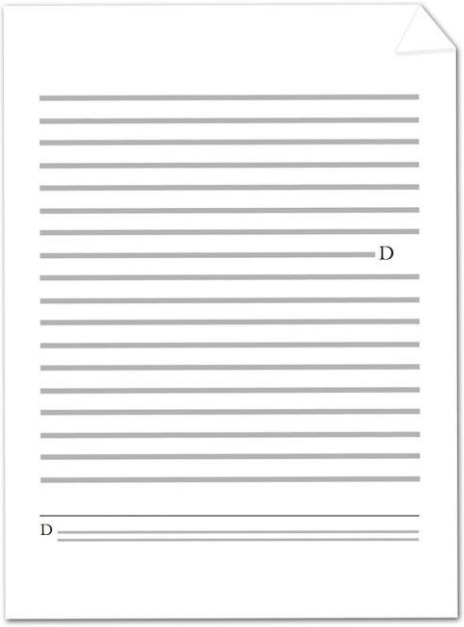
This element specifies the starting number or character for the first automatically numbered footnotes or endnote in the document, as well as the first automatically numbered footnotes after each restart point specified by the numRestart element (§17.11.19). This value shall be specified in decimal number units, then translated accordingly to the appropriate numbering format.

If this element is omitted, then the starting value shall be 1.

[*Example*: Consider the following footnote reference with the number format set to upper case letters and starting character set to D:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



The number format is specified by the following WordprocessingML:

```
<w:footnotePr>
  <w:numFmt w:val="upperLetter" />
  <w:numStart w:val="4" />
</w:footnotePr>
```

Since D is the fourth letter in the alphabet, the starting character is set to 4. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.11.21 pos (Footnote Placement)

This element specifies where footnotes shall be placed on the page when they are referenced by text in the current document.

If this element is present at the section level, then it shall be ignored.

If this element is omitted at the document level, then footnotes shall be located at the bottom of the current page.

[*Example*: Consider the following one page document, where the footnote appears beneath the text that it is referencing:

The footnote references the text reading `Some reference text.` and is represented by the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
     <w:t>Some referenced text</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:rStyle w:val="FootnoteReference" />
    </w:rPr>
    <w:footnoteReference w:id="2" />
  </w:r>
</w:p>
```

Since the footnote location must be beneath the current text, the section properties must be declared as follows:

```
<w:sectPr>
  …
  <w:footnotePr>
    <w:pos w:val="beneathText" />
  </w:footnotePr>
  …
</w:sectPr>
```

The footnote references the footnote in the footnotes part with an id attribute value equal to 2. Within the section properties of the document, the position of footnotes is specified to be beneath the page's text. *end example*]

Attributes	Description

val (Footnote Position Type)	Specifies the position of footnotes in the document.
	
	[*Example*: Consider a document in which footnotes must be positioned beneath their text. The footnote properties for this document must be declared as follows:
	
	`<w:sectPr>`
	`  <w:footnotePr>`
	`  <w:pos w:val="beneathText" />`
	`  </w:footnotePr>`
	`  …`
	`</w:sectPr>`
	
	The val attribute is beneathText, therefore the position of footnotes is specified to be beneath the page's text. *end example*]
	
	The possible values for this attribute are defined by the ST_FtnPos simple type (§17.18.34).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_FtnPos) is located in §A.1. *end note*]

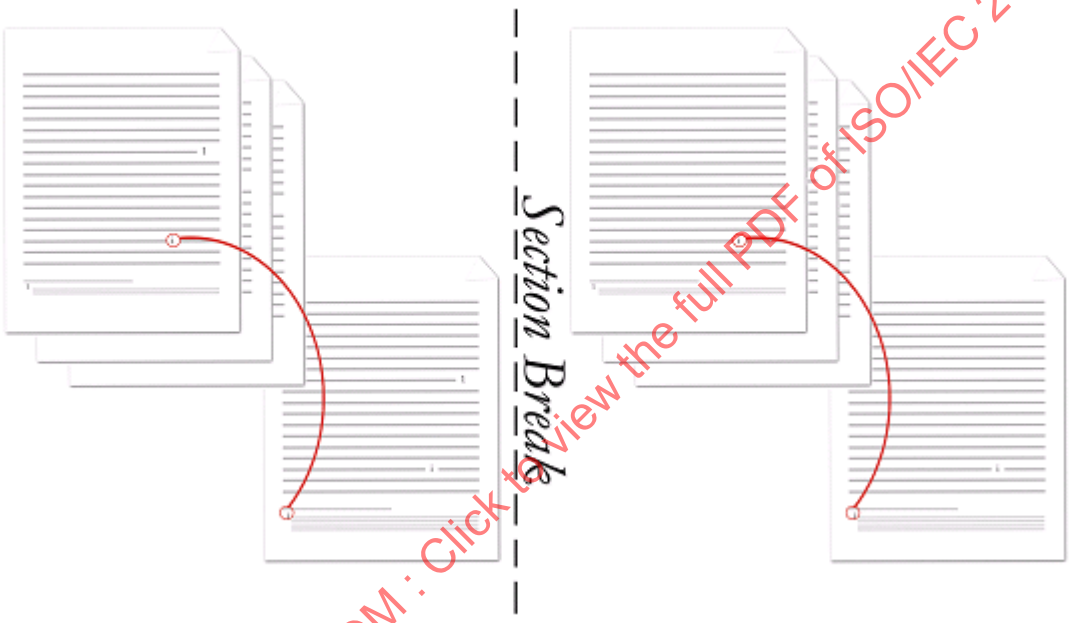
17.11.22 pos (Endnote Placement)

This element specifies where endnotes shall be placed on the page when they are referenced by text in the current document.

If this element is present at the section level, then it shall be ignored.

If this element is omitted at the document level, then endnotes shall be located at the end of the document.

[*Example*: Consider the following multi-page, multi-section document, where the endnote appears at the end of the section in which it is referenced:



The endnote setting is the same for all sections, and is represented by the following WordprocessingML at the document level:

```
<w:settings>
  …
  <w:endnotePr>
    <w:pos w:val="sectEnd" />
    …
  </w:endnotePr>
  …
</w:settings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Within the properties of the document, the position of endnotes is specified to be at the end of each section. *end example*]

Attributes	Description

val (Endnote Position Type)	Specifies the position of endnotes on the parent section or the document.
	
	[*Example*: Consider a document in which endnotes must be positioned at the end of the section. The section properties for this section must be declared as follows:
	
	`<w:settings>`
	`  <w:endnotePr>`
	`  <w:pos w:val="sectEnd" />`
	`  </w:endnotePr>`
	`  …`
	`</w:settings>`
	
	The val attribute is `endSect`, therefore the position of endnotes is specified to be at the end of the section. *end example*]
	
	The possible values for this attribute are defined by the ST_EdnPos simple type (§17.18.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EdnPos) is located in §A.1. *end note*]

17.11.23 separator (Footnote/Endnote Separator Mark)

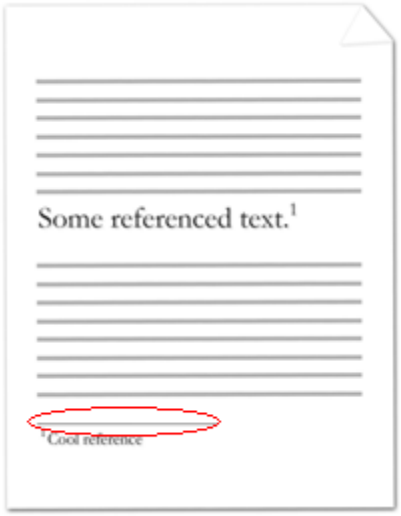
This element specifies the presence of a separator mark within the current run. A *separator mark* is a horizontal line which spans part of the width text extents.

[*Note*: The separator mark is typically used within the context of separator footnotes or endnotes. These footnote and endnote types define the footnote/endnote used to separate the contents of the main document story from the contents of footnotes or endnotes on that page. *end note*]

[*Example*: Consider the following page in a document, where some text is referenced by a footnote that is located at the bottom of the current page (with the separator circled in red):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



The line separating the document text from the normal footnotes is the footnote separator, and is represented by the following WordprocessingML:

```
<w:footnote w:type="separator" w:id="0">
  <w:p>
    <w:r>
      <w:separator />
    </w:r>
  </w:p>
</w:footnote>
```

In this example, the footnote has a content which consists of a single separator, which is displayed as a horizontal line across part of the text extents. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.12 Glossary Document

Within a WordprocessingML file, the *glossary document* is a supplemental storage location for additional document content which shall travel with the document, but which shall not be displayed for printed as part of the main document until it is explicitly added to that document by deliberate action.

The glossary document shall also be afforded a separate instance of all of the relationships which are provided on the main document part - this means that the glossary document shall have its own style definitions, numbering definitions, comments, headers, footers, etc. within the WordprocessingML document.

[*Example*: Consider a document which must include ten optional clauses that can be inserted through a user interface. It is clearly not desirable to have these ten clauses appear in the main document story's contents before they are explicitly inserted, therefore each of them can be stored in the glossary document and inserted via the user interface as needed. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Within the glossary document, each distinct region of document content is referred to as a *glossary document entry*, and is defined via the docPart element (§17.12.5). These document parts can contain any block-level WordprocessingML element, and can also have a set of classifications and behaviors applied to them via the glossary document entry's properties.

[*Example*: Consider the following definition for the contents of a glossary document part within a WordprocessingML document:

```
<w:glossaryDocument>
  <w:docParts>
    <w:docPart>
      <w:docPartPr>
        …
      </w:docPartPr>
      <w:docPartBody>
        <w:p>
          <w:r>
            <w:t>Sample entry.</w:t>
          </w:r>
        </w:p>
      </w:docPartBody>
    </w:docPart>

    <w:docPart>
      …
    </w:docPart>
  </w:docParts>
</w:glossaryDocument>
```

The glossaryDocument element (§17.12.10) defines the contents of the glossary document part. Within the glossary document, each docPart element contains the definition for one glossary document entry: in this case, there are two entries in the glossary document, the first of which contains a single paragraph with a single run of text. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.12.1 behavior (Entry Insertion Behavior)

This element specifies a single behavior which shall be applied to the contents of the parent glossary document entry (§17.12.5) when it is added to the main document story of a WordprocessingML document. These behaviors shall be used to format the surrounding WordprocessingML around insertion, and do not require the presence of a user interface (i.e. applications without a user interface shall also utilize these settings).

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

©ISO/IEC 2016 – All rights reserved

```
<w:docPart>
  <w:docPartPr>
    <w:behaviors>
      <w:behavior w:val="p"/>
    </w:behaviors>
    …
  </w:docPartPr>
  <w:docPartBody>
    <w:p>
      <w:r>
        <w:t>Sample entry.</w:t>
      </w:r>
    </w:p>
  </w:docPartBody>
</w:docPart>
```

The behavior element has a value of p, which specifies that the contents of the parent glossary document entry must be inserted in their own paragraph when they are added to the contents of a document. If the document content to which they are added is defined as follows (and the part is added between the two text runs):

```
<w:body>
  <w:p>
    <w:r>
      <w:t>After this text</w:t>
    </w:r>
    <w:r>
      <w:t>Before this text</w:t>
    </w:r>
  </w:p>
</w:body>
```

This setting specifies that although the part would normally be inserted between the two existing runs in the paragraph, that behavior must ensure that the part is inserted into its own paragraph, resulting in the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:body>
  <w:p>
    <w:r>
      <w:t>After this text</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t>Sample entry.</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t>Before this text</w:t>
    </w:r>
  </w:p>
</w:body>
```

end example]

Attributes	Description

val (Insertion Behavior Value)	Specifies the insertion behavior which shall be associated with the current glossary document entry.
	
	[*Example*: Consider the WordprocessingML fragment for a glossary document entry's properties, defined as follows:
	
	`<w:docPartPr>`
	`  <w:behaviors>`
	`  <w:behavior w:val="content"/>`
	`  </w:behaviors>`
	`</w:docPartPr>`
	
	The val attribute value of `content` specifies that the insertion of this glossary document entry must include only the content (the last paragraph in the part must be merged into the current paragraph in the document). *end example*]
	
	The possible values for this attribute are defined by the ST_DocPartBehavior simple type (§17.18.15).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartBehavior) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.12.2 behaviors (Entry Insertion Behaviors)

This element specifies the set of behaviors which shall be applied to the contents of the parent glossary document entry (§17.12.5) when it is added to the main document story of a WordprocessingML document. Since multiple behaviors can be specified for a single part, the sum total of all behaviors shall be used to insert the parent entry into the contents of the WordprocessingML document.

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

```
<w:docPart>
  <w:docPartPr>
    <w:behaviors>
      <w:behavior w:val="p" />
      <w:behavior w:val="pg" />
    </w:behaviors>
    …
  </w:docPartPr>
  <w:docPartBody>
    <w:p>
      <w:r>
        <w:t>Sample entry.</w:t>
      </w:r>
    </w:p>
  </w:docPartBody>
</w:docPart>
```

The behaviors element contains the set of behaviors which must be applied to this entry when it is inserted into the document, in this case:

- The entry must be inserted into its own paragraph in the document
- The entry must be inserted onto a new page in the document (i.e. it must be preceded by a page break)

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartBehaviors) is located in §A.1. *end note*]

17.12.3 category (Entry Categorization)

This element specifies the categorization for the parent glossary document entry. This categorization shall not imply any behaviors around the entry, and is only used to organize the set of glossary document entries within an application or user interface (i.e. to disambiguate between two entries with the same entry name (§17.12.13)).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:category>
    …
  </w:category>
  …
</w:docPartPr>
```

The category element specifies the categorization applied to the current entry, for the purposes of classification or user interface sorting, for example. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartCategory) is located in §A.1. *end note*]

17.12.4 description (Description for Entry)

This element specifies a description for the contents of this glossary document entry. This description can contain any string content, and allows the entry to have additional information contained within the definition for this glossary document entry. [*Note*: This description can be surfaced in a user interface, for example. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  …
  <w:name w:val="Sample Entry" />
  <w:description w:val="This is an example of a glossary document entry for
example purposes." />
  …
</w:docPartPr>
```

The description element specifies that the long description associated with the parent entry must be `This is an example of a glossary document entry for example purposes.` This value can be used as needed by an application, for example, to display in a user interface. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.12.5 docPart (Glossary Document Entry)

This element specifies the details for a single glossary document entry contained in the document. This glossary document entry can consist of one or both of the following:

- The glossary document entry's properties, which define its name, categorization, and behaviors
- The glossary document entry's contents, which consists of one or more block-level elements of WordprocessingML content

Each of these two components is specified by one of the child elements of this element, as seen in the child elements table below.

[*Example*: Consider the following definition for the contents of a Glossary Document part within a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:glossaryDocument>
  <w:docParts>
    <w:docPart>
      …
    </w:docPart>
    <w:docPart>
      …
    </w:docPart>
  </w:docParts>
</w:glossaryDocument>
```

The docPart element uniquely defines one glossary document entry within the glossary document part, therefore there are two unique entries stored in the current example of a glossary document part. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPart) is located in §A.1. *end note*]

17.12.6 docPartBody (Contents of Glossary Document Entry)

This element specifies the contents of the parent glossary document entry (§17.12.5). These contents shall consist of one or more block-level elements, analogous to the body element (§17.2.2) of the main document story for the current document.

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

```
<w:docPart>
  …
  <w:docPartBody>
    <w:p>
      <w:r>
        <w:t>Sample entry.</w:t>
      </w:r>
    </w:p>
  </w:docPartBody>
</w:docPart>
```

The docPartBody element specifies the block-level elements which comprise the contents of the current glossary document entry, in this case, a single paragraph using the p element (§17.3.1.22). *end example*]

When the contents of a glossary document entry are added to a document, the styles, numbering definitions, and all other related parts for this entry shall be taken from the relationships from the Glossary Document part and not from the main document part. These references shall be moved to their main document equivalents when the entry is added to the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

When the part is inserted, it shall be inserted as though its last paragraph mark does not exist (the content of the final paragraph mark shall be merged with the contents of the paragraph into which this entry is being added).

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

```
<w:docPart>
  <w:docPartPr>
    <w:behaviors>
      <w:behavior w:val="p"/>
    </w:behaviors>
    …
  </w:docPartPr>
  <w:docPartBody>
    <w:p>
      <w:r>
        <w:t>Sample entry.</w:t>
      </w:r>
    </w:p>
  </w:docPartBody>
</w:docPart>
```

If this entry is inserted into document content to which is defined as follows (and the part is added between the two text runs):

```
<w:body>
  <w:p>
    <w:r>
      <w:t>After this text</w:t>
    </w:r>
    <w:r>
      <w:t>Before this text</w:t>
    </w:r>
  </w:p>
</w:body>
```

This entry has only a single paragraph, which is removed before insertion, and barring any special insertion behaviors (§17.12.2), only the text run is inserted, resulting in the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:body>
  <w:p>
    <w:r>
      <w:t>After this text</w:t>
    </w:r>
    <w:r>
      <w:t>Sample entry.</w:t>
    </w:r>
    <w:r>
      <w:t>Before this text</w:t>
    </w:r>
  </w:p>
</w:body>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Body) is located in §A.1. *end note*]

17.12.7 docPartPr (Glossary Document Entry Properties)

This element specifies the set of properties which shall be applied to the parent glossary document entry. These properties define its name, categorization, and behaviors.

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

```
<w:docPart>
  <w:docPartPr>
    <w:name w:val="Sample Entry" />
    …
  </w:docPartPr>
  …
</w:docPart>
```

The docPartPr element specifies the set of properties which have been specified for the parent glossary document entry, the only one visible above being the entry's name of `Sample Entry`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartPr) is located in §A.1. *end note*]

17.12.8 docParts (List of Glossary Document Entries)

This element specifies the collection of glossary document entries which are stored in the current Glossary Document part.

[*Example*: Consider the following definition for the contents of a glossary document part within a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:glossaryDocument>
  <w:docParts>
    <w:docPart>
      …
    </w:docPart>
    <w:docPart>
      …
    </w:docPart>
  </w:docParts>
</w:glossaryDocument>
```

The docParts element defines the set of entries which are stored in the glossary document part. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocParts) is located in §A.1. *end note*]

17.12.9 gallery (Gallery Associated With Entry)

This element specifies the predefined gallery into which the current glossary document part shall be classified. This classification, although its enumeration values can be interpreted to imply semantics around the contents of the parent glossary document entry, shall only be used to classify and sort this entry (via an application or a user interface).

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:category>
    <w:name w:val="Internal Memo Covers" />
    <w:gallery w:val="coverPg" />
  </w:category>
  …
</w:docPartPr>
```

The gallery element with a value of coverPg specifies that the gallery categorization applied to the current entry, for the purposes of classification or user interface sorting, puts this entry into the Cover Pages classification. *end example*]

Attributes	Description
val (Gallery Value)	Specifies the classification of gallery which shall be associated with the parent glossary document entry. [*Example*: Consider the following WordprocessingML fragment for a single glossary document entry: `<w:gallery w:val="custom1" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The val attribute with a value of `custom1` specifies that the gallery categorization applied to the current entry is the `Custom 1` classification. *end example*] The possible values for this attribute are defined by the ST_DocPartGallery simple type (§17.18.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartGallery) is located in §A.1. *end note*]

17.12.10 glossaryDocument (Glossary Document Root Element)

This element specifies the root element for a glossary document part within a WordprocessingML document. A glossary document is an supplementary document story in a WordprocessingML that shall be afforded all of the relationships of the Main Document part, such as:

- Style definitions
- Numbering definitions
- Comments
- Headers/footers
- Etc.

The entries stored in this part shall have all of its implicit relationships target these parts, rather than their analogues stored off of the main document part.

[*Example*: Consider the following definition for the contents of a glossary document part within a WordprocessingML document:

```
<w:glossaryDocument>
  <w:docParts>
    <w:docPart>
      …
    </w:docPart>
    <w:docPart>
      …
    </w:docPart>
  </w:docParts>
</w:glossaryDocument>
```

The glossaryDocument element defines the contents of the glossary document part. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GlossaryDocument) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.12.11 guid (Entry ID)

This element specifies a unique identifier (specified using a 128-bit GUID stored on the val attribute) that uniquely identifies this document building block.[*Note*: This unique identifier can be used by an application to uniquely reference a single document building block regardless of different naming, for example when the same part has different names for localization purposes. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  …
  <w:guid w:val="{00000000-5BD2-4BC8-9F70-7020E1357FB2}" />
  …
</w:docPartPr>
```

The guid element specifies that the unique identifier associated with the parent entry must be {00000000-5BD2-4BC8-9F70-7020E1357FB2}. This value can be used as needed by an application, for example, to uniquely identify a part regardless of its name. *end example*]

Attributes	Description
val (GUID Value)	Specifies a 128-bit globally unique identifier (GUID) value as defined by the simple type referenced below. The contents of this GUID shall be interpreted based on the context of the parent XML element. If this attribute is omitted, its value shall be assumed to be null (i.e. no GUID shall be associated with the parent XML element). [*Example*: Consider the following WordprocessingML element: <… w:val="{6A9B8B6F-5BD2-4BC8-9F70-7020E1357FB2}"/> The val attribute value of {6A9B8B6F-5BD2-4BC8-9F70-7020E1357FB2} must be associated with the context of the parent XML element. *end example*] The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Guid) is located in §A.1. *end note*]

17.12.12 name (Category Associated With Entry)

This element specifies the category into which the current glossary document part shall be classified. This classification can consist of any string value as determined by its contents, and shall only be used to classify and sort this entry (via an application or a user interface).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:category>
    <w:name w:val="Internal Memo Covers" />
    <w:gallery w:val="coverPg" />
  </w:category>
  …
</w:docPartPr>
```

The name element with a value of `Internal Memo Covers` specifies that the category grouping applied to the current entry, for the purposes of classification or user interface sorting, puts this entry into the `Internal Memo Covers` classification. This category can be used as desired. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.12.13 name (Entry Name)

This element specifies a name for the contents of this glossary document entry. This name can contain any string content, and allows the entry to have a friendly identifier contained within the definition for this glossary document entry. [*Note*: This name can be surfaced in a user interface, for example. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  …
  <w:name w:val="Sample Entry" />
  <w:description w:val="This is an example of a glossary document entry for
example purposes." />
  …
</w:docPartPr>
```

The name element specifies that the friendly name associated with the parent entry must be `Sample Entry`. This value can be used as needed by an application, for example, to display in a user interface. *end example*]

Attributes	Description
decorated (Built-In Entry)	Specifies that the name for the current entry is a built-in entry which should not be displayed in the user interface. [*Note*: This information can be used by an application as needed, for example, to disambiguate an entry from one with the same name, ensuring that the built-in entry can be uniquely identified by the application. *end note*] If this attribute is omitted, its value shall be assumed to be `false`. [*Example*: Consider the following WordprocessingML fragment for the name of a single glossary document entry: `<w:name w:decorated="true" w:val=":-)" />` The decorated attribute specifies that the parent entry is a built-in entry, and must be treated as such. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
val (Name Value)	Specifies a string value which contains the name of the current glossary document entry. [*Example*: Consider the following WordprocessingML fragment for the name of a single glossary document entry: `<w:name w:val="Sample Entry" />` The val attribute specifies that the name of the parent entry is `Sample Entry`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartName) is located in §A.1. *end note*]

17.12.14 style (Associated Paragraph Style Name)

This element specifies the style ID for a paragraph style which shall be associated with the current glossary document entry. This paragraph style associated shall not imply anything about the formatting or content of the glossary document entry, and shall only be used to filter and/or sort this entry (via an application or a user interface). [*Note*: One example of the level of classification offered by this element is to only show it as available when the formatting of the paragraph matches the specified style. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:style w:val="Heading1" />
  …
</w:docPartPr>
```

The style element with a val attribute value of `Heading1` specifies that the paragraph style associated with the current glossary document entry must be the style whose style ID is equal to `Heading1`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.12.15 type (Entry Type)

This element specifies an entry type which shall be applied to the properties of the parent glossary document entry (§17.12.5). Each of these entry types can, based on their values, influence the visibility and behavior of the parent glossary document entry as defined by the associated simple type information.

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:types>
    <w:type w:val="bbPlcHdr" />
  </w:types>
  …
</w:docPartPr>
```

The type element with a value of bbPlcHdr specifies that the parent glossary document entry must be treated as if it was the placeholder text for one or more structured document tags in the document. *end example*]

Attributes	Description
val (Type Value)	Specifies the value for the current entry type. [*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry: <w:type w:val="bbPlcHdr" /> The val attribute value of bbPlcHdr specifies that the parent glossary document entry must be treated as if it was the placeholder text for one or more structured document tags in the document. *end example*] The possible values for this attribute are defined by the ST_DocPartType simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartType) is located in §A.1. *end note*]

17.12.16 types (Entry Types)

This element specifies the set of entry types which shall be applied to the properties of the parent glossary document entry (§17.12.5). Each of these entry types can, based on their values, influence the visibility and behavior of the parent glossary document entry.

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:types>
    …
  </w:types>
  …
</w:docPartPr>
```

The types element specifies the set of entry types which must be associated with the parent glossary document entry. *end example*]

Attributes	Description

all (Entry Is Of All Types)	Specifies that the current glossary document is all entry types. This attribute shall override any information specified as child elements of this element and shall ensure that the current entry is associated with all available entry types.
	
	If this attribute is omitted, then its default value shall be assumed to be `false`.
	
	[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:
	
	`<w:docPartPr>`
	`  <w:types w:all="true">`
	`  <w:type w:val="autoExp" />`
	`  </w:types>`
	`  …`
	`</w:docPartPr>`
	
	The types element contains a single entry type definition, but because the all attribute is present with a value of `true`, that entry type is augmented to place the parent entry into all possible entry types. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocPartTypes) is located in §A.1. *end note*]

17.13 Annotations

Within a WordprocessingML document, *annotations* refer to various types of supplementary markup which can be stored inside or around a region of text within the document's contents. [*Example*: The types of supplementary information stored within a document can include: comments, revisions, spelling and/or grammatical errors, bookmark information and optional editing permissions. *end example*]

Within a document's contents, annotations are stored in one of three different methods:

- Inline
- "Cross Structure"
- Properties

These three forms are needed in order to maintain compatibility with both the legacy annotations functionality of current word processing applications and the requirements of an XML-based format (i.e. wellformedness of the resulting XML markup). These three forms are referenced within the individual annotation types described in the following sub clauses.

17.13.1 Inline Annotations

Inline annotations describe all annotations which do not require special handling in order to maintain the XML wellformedness requirements of the resulting WordprocessingML output. In these cases, a single XML element shall encapsulate the entire contents of the document content which is being annotated.

[*Example*: Consider the following WordprocessingML markup for a paragraph which reads `The quick brown fox jumps over the jet lagged dog.`, where `jet lagged` replaced the previous text `lazy` when the editing application was tracking revisions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">The quick brown fox jumps over the </w:t>
  </w:r>
  <w:del … >
    <w:r>
      <w:delText>lazy</w:delText>
    </w:r>
  </w:del>
  <w:ins … >
    <w:r>
      <w:t>jet lagged</w:t>
    </w:r>
  </w:ins>
  <w:r>
    <w:t xml:space="preserve"> dog.</w:t>
  </w:r>
</w:p>
```

The del and ins elements (§17.13.5.14; §17.13.5.18) each fully encapsulate the extent of their respective annotations (a marked deletion and insertion, respectively), as they are inline annotations. *end example*]

17.13.2 "Cross Structure" Annotations

"Cross structure" annotations describe the class of annotations which can span portions of WordprocessingML markup [*Example*: Cross structure annotations can span parts of multiple paragraphs, one half of a custom XML markup element's contents, etc. *end example*]. In these cases, the annotation's region is delimited by two elements: a start element and an end element. These two elements mark the start and end points of the annotated content, but do not contain it. The pairing of the start and end marker are linked via a common value for their id attributes.

[*Example*: Consider the following WordprocessingML markup for two paragraphs, each reading `Example Text`, where a bookmark has been added spanning the second word in paragraph one and the first word in paragraph two:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t>Example</w:t>
  </w:r>
  <w:bookmarkStart w:id="0" w:name="sampleBookmark" />
  <w:r>
    <w:t xml:space="preserve"> text.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>Example</w:t>
  </w:r>
  <w:bookmarkEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve"> text.</w:t>
  </w:r>
</w:p>
```

The bookmarkStart and bookmarkEnd elements (§17.13.6.2; §17.13.6.1) specify the location where the bookmark starts and ends, but cannot contain it because it spans part of two paragraphs. They are part of one group because the id attribute value specifies `0` for both. *end example*]

17.13.3 Property Annotations

Property annotations describe the class of annotations which are stored as a property on an object [*Example*: Property annotations can appear on paragraph properties, run properties, table rows, etc. *end example*] In these cases, the annotation's semantics are defined by the property, as they can affect content and/or formatting.

[*Example*: Consider the following WordprocessingML markup for a paragraph reading `Example Text`, where the first word had the bold property applied when the editing application was tracking revisions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:rPr>
      <w:b/>
      <w:rPrChange … >
        <w:rPr/>
      </w:rPrChange>
    </w:rPr>
    <w:t>Example</w:t>
  </w:r>
  <w:r>
    <w:t xml:space="preserve"> text.</w:t>
  </w:r>
</w:p>
```

The rPrChange element (§17.13.5.31; §17.13.5.30) contains the set of previously applied revision properties associated with a particular author at a particular time. It is stored itself as a property on the parent run which was modified. *end example*]

17.13.4 Comments

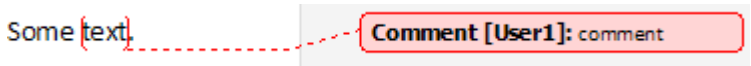
Comments describe annotations which are anchored to a region of document content, but which contain an arbitrary amount of block-level content stored in their own separate document stories. Within a WordprocessingML document, comments are stored in a separate Comments part within the document package.

A comment in a WordprocessingML document is divided into two components:

- The comment anchor (the text on which the comment applies)
- The comment content (the contents of the comment)

The *comment anchor* is the cross structure annotation which defines the region of text on which the comment in anchored. The *comment content* is the content stored in the comments part which contains the actual content of the comment.

[*Example*: Consider a paragraph in a WordprocessingML document whose second word is annotated with a comment:



The first component to this comment is the document content which defines the extents of the comment and references the specific comment in the comments part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Some </w:t>
  </w:r>
  <w:commentRangeStart w:id="0" />
  <w:r>
    <w:t>text.</w:t>
  </w:r>
  <w:commentRangeEnd w:id="0" />
  <w:r>
    <w:commentReference w:id="0" />
  </w:r>
</w:p>
```

The commentRangeStart and commentRangeEnd elements (§17.13.4.4; §17.13.4.3) delimit the run content to which the comment with an id of `0` applies (in this case, the single run of text). The following commentReference element (§17.13.4.5) links the preceding run content with a comment in the comments part with an id of `0`. Without all three of these elements, the range and comment cannot be linked (although the first two elements are optional, in which case the comment must be anchored at the comment reference mark)

The second component to this comment is the comment content which defines the text in the comment:

```
<w:comment w:id="0" w:author="Joe Smith" w:date="2006-04-06T13:50:00Z"
w:initials="User">
  <w:p>
    <w:pPr>
      <w:pStyle w:val="CommentText" />
    </w:pPr>
    <w:r>
      <w:rPr>
        <w:rStyle w:val="CommentReference" />
      </w:rPr>
      <w:annotationRef />
    </w:r>
    <w:r>
      <w:t>comment</w:t>
    </w:r>
  </w:p>
</w:comment>
```

In this example, the comment specifies that it was inserted by author `Joe Smith` with the initials `User` via the author and date attributes. It is linked to the run content via the id attribute, which matches the value of `0` specified using the commentReference element above. The block-level content of the comment specifies that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

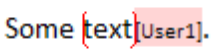
its text is `comment` and the style of the comment content is based off of the character style with the name `CommentReference`. *end example*]

17.13.4.1 annotationRef (Comment Information Block)

This element specifies the presence of an annotation reference mark at the current location in the comment. An *annotation reference mark* is an information block that represents the metadata about the current comment within the document. This annotation reference mark should typically consist of the initials and a unique integer associated with its position in the document, but can be displayed in any desired format.

If this element is omitted from a single comment's contents, then an annotation reference mark can be added at the start of the comment in reading order (right in a right-to-left paragraph or left in a left-to-right paragraph). As well, an annotation reference mark can be relocated as desired within a comment's content.

[*Example*: Consider a document with text with an annotated comment as follows:



This comment is represented as the following WordprocessingML fragment:

```
<w:comment … w:initials="User">
  <w:p>
    <w:r>
      <w:annotationRef />
    </w:r>
    …
  </w:p>
</w:comment>
```

The annotationRef element specifies that the comment must start with an annotation reference mark. In this example, this mark is displayed as a combination of the user initial, `User`, and a unique sequential number, `1`. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.13.4.2 comment (Comment Content)

This element specifies the content of a single comment stored in the Comments part of a WordprocessingML document.

If a comment is not referenced by document content via a matching id attribute on a correct use of the commentReference element (§17.13.4.5), as defined by the WordprocessingML schema, then it can be ignored when loading the document. If more than one comment shares the same value for the id attribute, then only one comment shall be loaded and the others can be ignored.

[*Example*: Consider a document with text with an annotated comment as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Some text[User1].

This comment is represented as the following WordprocessingML fragment:

```
<w:comment w:id="1" w:initials="User" w:author="Dan Battagin">
  …
</w:comment>
```

The comment element specifies the presence of a single comment within the comments part. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`  …`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

initials (Initials of Comment Author)	Specifies the initials of the author of the current comment. [*Note*: This information can be used to format and present the associated comment information block (§17.13.4.1), or in any user interface supported by an application. If there is more than one author with the same initials, it might be more useful to display the author name. *end note*]
	
	If this attribute is omitted, then no author shall be associated with the current comment in the document.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<w:comment w:id="1" w:initials="KB" w:author="Krista Bendig">`
	`…`
	`</w:comment>`
	
	The initials attribute specifies that the initials of the author of the current comment are KB, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Comment) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

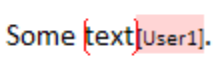
17.13.4.3 commentRangeEnd (Comment Anchor Range End)

This element specifies the end of the range around which a comment is anchored in the content of the WordprocessingML document. The id attribute on this element shall be used to link the corresponding comment anchor range start element and comment reference.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding commentRangeStart element (§17.13.4.4) with a matching id attribute value, then it shall be considered the single anchor point for the associated comment reference.
- If this element appears without a corresponding commentReference element (§17.13.4.5) in the current document story with a matching id attribute value, then the document is non-conformant.
- If this element appears in a comment content story (§17.13.4.2), then the document is non-conformant .

[*Example*: Consider a paragraph in a WordprocessingML document whose second word is annotated with a comment:



The WordprocessingML fragment for this comment is defined as follows:

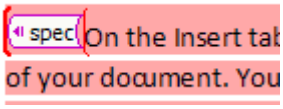
```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Some </w:t>
  </w:r>
  <w:commentRangeStart w:id="0" />
  <w:r>
    <w:t>text.</w:t>
  </w:r>
  <w:commentRangeEnd w:id="0" />
  <w:r>
    <w:commentReference w:id="0" />
  </w:r>
</w:p>
```

The commentRangeEnd element specifies that the end of the comment range for the comment with an id of `0` is after the end of the run containing the word `text`. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Markup)	custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkupRange) is located in §A.1. *end note*]

17.13.4.4 commentRangeStart (Comment Anchor Range Start)

This element specifies the start of the range around which a comment is anchored in the content of the WordprocessingML document. The id attribute on this element shall be used to link the corresponding comment anchor range end element and comment reference.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding commentRangeEnd element (§17.13.4.3) with a matching id attribute value, then it shall be considered the single anchor point for the associated comment reference.
- If this element appears without a corresponding commentReference element (§17.13.4.5) in the current document story with a matching id attribute value, then the document is non-conformant .
- If this element appears in a comment content story (§17.13.4.2), then the document is non-conformant .

[*Example*: Consider a paragraph in a WordprocessingML document whose second word is annotated with a comment:

Some text[User1].

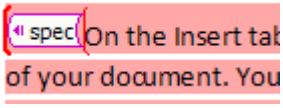
The WordprocessingML fragment for this comment is defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Some </w:t>
  </w:r>
  <w:commentRangeStart w:id="0" />
  <w:r>
    <w:t>text.</w:t>
  </w:r>
  <w:commentRangeEnd w:id="0" />
  <w:r>
    <w:commentReference w:id="0" />
  </w:r>
</w:p>
```

The commentRangeStart element specifies that the start of the comment range for the comment with an id of `0` is after the end of the run containing the word Some. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkupRange) is located in §A.1. *end note*]

17.13.4.5 commentReference (Comment Content Reference Mark)

This element specifies the presence of a comment content reference mark, which links the comment content (§17.13.4.2) with the contents of a document story. This link is established by matching the comment whose id

©ISO/IEC 2016 – All rights reserved

attribute matches the id attribute on this element. The resulting comment is anchored to the range with comment range elements with the same id attribute values (if present) as follows:

- If both of the commentRangeStart and commentRangeEnd elements (§17.13.4.4; §17.13.4.3) are present, then the comment reference shall anchor the comment to the resulting range.
- If only one of the commentRangeStart and commentRangeEnd elements (§17.13.4.4; §17.13.4.3) is present, then the document is non-conformant.
- If neither element is present, then the comment reference shall anchor the comment to its current location.

If this element appears in a comment content story (§17.13.4.2), then it can be ignored. If no comment exists with an id attribute which matches the id attribute on this element, then this document is non-conformant.

[*Example*: Consider a paragraph in a WordprocessingML document whose second word is annotated with a comment:

Some text[User1].

The WordprocessingML fragment for this comment is defined as follows:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Some </w:t>
  </w:r>
  <w:commentRangeStart w:id="0" />
  <w:r>
    <w:t>text.</w:t>
  </w:r>
  <w:commentRangeEnd w:id="0" />
  <w:r>
    <w:commentReference w:id="0" />
  </w:r>
</w:p>
```

The commentReference element specifies that the associated comment in the comments part must be the comment whose id attribute value is 0. As well, since a start and end marker exist with a matching ID, this comment is anchored to that region of the document. *end example*]

Attributes	Description
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Markup) is located in §A.1. *end note*]

17.13.4.6 comments (Comments Collection)

This element specifies all of the comments defined in the current document. It is the root element of the Comments part of a WordprocessingML document.

[*Example*: Consider the following WordprocessingML fragment for the content of a comments part in a WordprocessingML document:

```
<w:comments>
  <w:comment … >
    …
  </w:comment>
</w:comments>
```

The comments element contains the single comment specified by this document in this example. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Comments) is located in §A.1. *end note*]

17.13.5 Revisions

Revisions in WordprocessingML provide a mechanism for storing information about the evolution of the document (i.e. the set of modifications made to a document by one of more authors). When an application adds revisions to the content of a WordprocessingML document, they are specifying this by storing either (depending on the revision type):

- The current state of the document (a deletion stores the current state of the text as deleted, and implies that its original state was the content used to exist)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- The initial state of the document (a run's initial properties are explicitly stored in a previous run properties block, as the current run properties are always those that are the child of the rPr element (§17.7.9.1))

A revision consists of two required pieces of information:

- The revision type (specified via the name of the revision element)
- A unique revision identifier (used to uniquely identify revisions)

As well as optional information:

- The author of the revision
- The date and time of the revision

[*Example*: Consider a paragraph of text in a WordprocessingML document in which one word has been inserted, as follows:

Some text

This paragraph has the word `text` marked inserted as a revision, and is represented as the following WordprocessingML:

```
<w:p>
  <w:r>
    <w:t>Some</w:t>
  </w:r>
  <w:ins w:id="0" w:author="Joe Smith" w:date="2006-03-31T12:50:00Z">
    <w:r>
      <w:t>text</w:t>
    </w:r>
  </w:ins>
</w:p>
```

The ins element contains both the required information: all of the content which must be treated as revision marked as inserted (the word `text`); a unique revision identifier of `0`.

The element also stores the optional information about the revision: the word `text` was inserted by `Joe Smith` on March 31, 2006 at 12:50pm. *end example*]

Within a WordprocessingML document, the following types of revisions can be used to track the changes to a document:

- Insertions
- Deletions
- Moves

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- Changes to run/paragraph/table/numbering/section properties
- Changes to custom XML markup

17.13.5.1 cellDel (Table Cell Deletion)

This element specifies that the parent table cell shall be treated as though it was deleted from the document while revisions were being recorded. This means that although the table cell element exists in the structure of the table, the table cell technically no longer exists in the document.

[*Example*: Consider a document with a two row by two columns table as follows:

One	Two
Three	Four

If this table has each cell in its final column deleted and this is tracked as a revision, the resulting WordprocessingML would show each of these cells as deleted as follows:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>One</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:cellDel w:id="0" … />
      </w:tcPr>
      <w:r>
        <w:t>Two</w:t>
      </w:r>
    </w:tc>
  </w:tr>
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>Three</w:t>
      </w:r>
    </w:tc>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <w:tc>
        <w:tcPr>
          <w:cellDel w:id="1" … />
        </w:tcPr>
        <w:r>
          <w:t>Four</w:t>
        </w:r>
      </w:tc>
    </w:tr>
  </w:tbl>
```

The cellDel elements in the table cell properties of the cells with text Two and Four specify that each of those cells have been deleted from the document. Their attributes (omitted) can optionally provide information about the time at which this deletion took place. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`  …`
	`</…>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.2 cellIns (Table Cell Insertion)

This element specifies that the parent table cell shall be treated as though it was inserted into the document while revisions were being recorded.

[*Example*: Consider a document with a two row by two columns table as follows:

One	Two
Three	Four

If this table has two cells added by appending an additional column, and this is tracked as a revision, the resulting WordprocessingML would show each of these cells as inserted as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>One</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:r>
        <w:t>Two</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:cellIns w:id="0" … />
      </w:tcPr>
      <w:r>
        <w:t>New</w:t>
      </w:r>
    </w:tc>
  </w:tr>
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>Three</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:r>
        <w:t>Four</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:cellIns w:id="1" … />
      </w:tcPr>
      <w:r>
        <w:t>New</w:t>
      </w:r>
    </w:tc>
  </w:tr>
</w:tbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The cellIns elements in the table cell properties of the cells with text New specify that each of those cells have been inserted into the document. Their attributes (omitted) can optionally provide information about the insertion of these cells (author, date, etc.). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.3 cellMerge (Vertically Merged/Split Table Cells)

This element specifies that the vertical merge state of the parent table cell has been modified while revisions were being tracked for the document. The vmerge and vmergeOrig attributes on this element specify the original and revised vertical merge states of the table cell.

[*Example*: Consider a document with a two row by two columns table as follows:

One	Two
Three	Four

If this table has the two cells in the second column merged into one and this modification is tracked as a revision, as follows:

One	Two
Three	Four

The resulting WordprocessingML for the revision would appear as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>One</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:r>
        <w:t>Two</w:t>
      </w:r>
    </w:tc>
  </w:tr>
  <w:tr>
    <w:tc>
      <w:r>
        <w:t>Three</w:t>
      </w:r>
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:cellMerge w:id="0" w:vMerge="cont"/>
      </w:tcPr>
      <w:r>
        <w:t>Four</w:t>
      </w:r>
    </w:tc>
  </w:tr>
</w:tbl>
```

The cellMerge element specifies that changes were made to the vertical merge settings of the last cell in the table, specifically; the cell was vertically merged with the cell above it (gaining a revised vmerge attribute value of cont). *end example*]

Attributes	Description
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document. If this attribute is omitted, then no author shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… w:id="1" w:author="Example Author"> … </…> The author attribute specifies that the author of the current annotation is Example Author, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
vMerge (Revised	Specifies the vertical merge setting which was applied to the parent table cell by this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

Vertical Merge Setting)	revision.
	
	If this attribute is omitted, then no revised vertical merge setting is supplied for this revision (if neither this nor the vmergeOrig attribute is specified, the revision can be ignored).
	
	[*Example*: Consider a two row by two column table in which the cells in the second column are merged, and this change is tracked as a revision. The annotation on the last cell in the table would appear as follows:
	
	`<w:tc>`
	`  <w:tcPr>`
	`  <w:cellMerge … w:vMerge="cont" />`
	`  </w:tcPr>`
	`  …`
	`</w:tc>`
	
	The vmerge attribute value of `cont` specifies that the revision on the table cell resulted in it being merged with the previous set of vertically merged cells above it (whether that was one cell or many). *end example*]
	
	The possible values for this attribute are defined by the ST_AnnotationVMerge simple type (§17.18.1).

vMergeOrig (Vertical Merge Setting Removed by Revision)	Specifies the vertical merge setting which was removed from the parent table cell by this revision.
	
	If this attribute is omitted, then the original vertical merge setting shall be assumed to be `rest` (not merged).
	
	[*Example*: Consider a two row by two column table in which the merged cells in the second column are split, and this change is tracked as a revision. The annotation on the last cell in the table would appear as follows:
	
	`<w:tc>`
	`  <w:tcPr>`
	`  <w:cellMerge … w:vMergeOrig="cont" />`
	`  </w:tcPr>`
	`  …`
	`</w:tc>`
	
	The vmergeOrig attribute value of `cont` specifies that the revision on the table cell resulted in it having its vertical merge property removed. *end example*]
	
	The possible values for this attribute are defined by the ST_AnnotationVMerge simple type (§17.18.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellMergeTrackChange) is located in §A.1. *end note*]

17.13.5.4 customXmlDelRangeEnd (Custom XML Markup Deletion End)

This element specifies the end of a region in which custom XML markup has been deleted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup deletion start marker in the document.

Providing a physical representation of custom XML markup results in regions which can be inserted and deleted, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlDelRangeStart element (§17.13.5.5) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant .
- If multiple end elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a document with two inline custom XML markup elements, as follows:

```
<w:p>
  <w:customXml … >
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

Now, if each custom XML markup element's start and end tag have a physical representation, imagine that the region from the start of the paragraph until the point between the two end points is deleted with revisions enabled. This revision cannot be encapsulated by one del element, since it starts outside of the first custom XML markup element and ends just inside of it, so it must be done using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:customXmlDelRangeStart w:id="0" w:author="Devan" />
<w:customXml … >
  <w:customXml … >
    <w:del … >
      <w:r>
        <w:delText>Text.</w:delText>
      </w:r>
    </w:del>
  </w:customXml>
  <w:customXmlDelRangeEnd w:id="0" />
</w:customXml>
<w:r>
  <w:t>More text.</w:t>
</w:r>
</w:p>
```

The customXmlDelRangeEnd element delimits the end of the region in which all custom XML elements have been deleted with revisions enabled, and the del element (§17.13.5.14) handles the deletion of the text performed by this revision. Since the end of the outer customXml element was not in the deleted range, it is not revision marked deleted, but the corresponding physical character for the start element is. *end example*]

Attributes	Description

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Markup) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.5 customXmlDelRangeStart (Custom XML Markup Deletion Start)

This element specifies the beginning of a region in which all custom XML markup has been deleted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup deletion end marker in the document.

Providing a physical representation of custom XML markup results in regions which can be inserted and deleted, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlDelRangeEnd element (§17.13.5.4) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired end encapsulate a range with no custom XML markup, then the document is non-conformant .
- If multiple start elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a document with two inline custom XML markup elements, as follows:

```
<w:p>
  <w:customXml … >
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

Now, if each custom XML markup element's start and end tag have a physical representation, imagine that the region from the start of the paragraph until the point between the two end points is deleted with revisions enabled. This revision cannot be encapsulated by one del element, since it starts outside of the first custom XML markup element and ends just inside of it, so it must be done using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:customXmlDelRangeStart w:id="0" w:author="Jamie" />
<w:customXml … >
  <w:customXml … >
    <w:del … >
      <w:r>
        <w:delText>Text.</w:delText>
      </w:r>
    </w:del>
  </w:customXml>
  <w:customXmlDelRangeEnd w:id="0" />
</w:customXml>
<w:r>
  <w:t>More text.</w:t>
</w:r>
</w:p>
```

The customXmlDelRangeStart element delimits the start of the region in which all custom XML elements have been deleted with revisions enabled, and the del element (§17.13.5.14) handles the deletion of the text performed by this revision. Since the end of the outer customXml element was not in the deleted range, it is not revision marked deleted, but the corresponding physical character for the start element is. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.6 customXmlInsRangeEnd (Custom XML Markup Insertion End)

This element specifies the end of a region within which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this insertion applies.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlInsRangeStart element (§17.13.5.7) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant .
- If multiple end elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a document with two inline custom XML markup elements, as follows:

```
<w:p>
  <w:customXml … >
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

If each custom XML markup element's start and end tag have a physical representation, consider that the inner XML element (but not its content) is inserted with revisions enabled. This revision cannot be encapsulated by one ins element, since the text in the element is not an insertion, so it must be done using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:p>
  <w:customXml … >
    <w:customXmlInsRangeStart w:id="0" />
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
    <w:customXmlInsRangeEnd w:id="0" />
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The customXmlInsRangeEnd element delimits the end of the region in which all custom XML elements have been inserted with revisions enabled. Since this element only affects custom XML, the text is not revision marked inserted, but the corresponding physical characters for the custom XML element are. *end example*]

Attributes	Description

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Markup) is located in §A.1. *end note*]

17.13.5.7 customXmlInsRangeStart (Custom XML Markup Insertion Start)

This element specifies the beginning of a region in which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion end marker in the document.

Providing a physical representation of custom XML markup start and end tags results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlInsRangeEnd element (§17.13.5.6) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired end encapsulate a range with no custom XML markup, then the document is non-conformant .
- If multiple start elements exist with the same id attribute value, then the document is non-conformant .

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This element specifies the end of a region within which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this insertion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlInsRangeStart element (§17.13.5.7) with a matching id attribute value, then it shall be ignored and no insertions shall be present in the document.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and can be omitted when the document is subsequently saved.
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

[*Example*: Consider a document with two inline custom XML markup elements, as follows:

```
<w:p>
  <w:customXml … >
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

If each custom XML markup element's start and end tag have a physical representation, consider that the inner XML element (but not its content) is inserted with revisions enabled. This revision cannot be encapsulated by one ins element, since the text in the element is not an insertion, so it must be done using the custom XML markup revision "cross structure" syntax, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:customXml … >
    <w:customXmlInsRangeStart w:id="0" />
    <w:customXml … >
      <w:r>
        <w:t>Text.</w:t>
      </w:r>
    </w:customXml>
    <w:customXmlInsRangeEnd w:id="0" />
  </w:customXml>
  <w:r>
    <w:t>More text.</w:t>
  </w:r>
</w:p>
```

The customXmlInsRangeStart element delimits the start of the region in which all custom XML elements have been inserted with revisions enabled. Since this element only affects custom XML, the text is not revision marked inserted, but the corresponding physical characters for the custom XML element are. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.8 customXmlMoveFromRangeEnd (Custom XML Markup Move Source End)

This element specifies the end of a region within which all custom XML markup was moved to another location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move source start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move source applies.

The following restrictions shall be applied to this element:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- If this element occurs without a corresponding customXmlMoveFromRangeStart element (§17.13.5.9) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant .
- If this element and its paired start occur outside of a well-formed move source container (§17.13.5.24; §17.13.5.23) with a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple end elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a three-paragraph document with a single block-level custom XML markup element, as follows:

```
<w:body>
  <w:p/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:p/>
</w:body>
```

If the second paragraph is moved to the end of the document with revisions enabled. This revision must therefore be stored using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:body>
  <w:p/>
  <w:moveFromRangeStart w:id="0" w:name="move1" w:displacedByCustomXml="next"
w:author="Luna"/>
  <w:customXmlMoveFromRangeStart w:id="1"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveFromRangeEnd w:id="1"/>
  <w:moveFromRangeEnd w:id="0" w:displacedByCustomXml="prev"/>
  <w:p/>
  <w:moveToRangeStart w:id="2" w:name="move1" w:displacedByCustomXml="next"/>
  <w:customXmlMoveToRangeStart w:id="3"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveToRangeEnd w:id="3"/>
  <w:moveFromRangeEnd w:id="2" w:displacedByCustomXml="prev"/>
</w:body>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The customXmlMoveFromRangeEnd element delimits the end of the region in which all custom XML elements have been moved from this location with revisions enabled. Since this element only affects custom XML, any text in the region is not revision marked moved by this element when present, but the corresponding physical characters for the custom XML element are. *end example*]

Attributes	Description

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Markup) is located in §A.1. *end note*]

17.13.5.9 customXmlMoveFromRangeStart (Custom XML Markup Move Source Start)

This element specifies the start of a region within which all custom XML markup was moved to another location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move source end marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move source applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlMoveFromRangeStart element (§17.13.5.9) with a matching id attribute value, then the document is non-conformant .
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant .

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- If this element and its paired end occur outside of a well-formed move source container (§17.13.5.24; §17.13.5.23) with a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple start elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a three-paragraph document with a single block-level custom XML markup element, as follows:

```
<w:body>
  <w:p/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:p/>
</w:body>
```

If the second paragraph is moved to the end of the document with revisions enabled. This revision must therefore be stored using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:body>
  <w:p/>
  <w:moveFromRangeStart w:id="0" w:name="move1" w:displacedByCustomXml="next"
w:author="Luke"/>
  <w:customXmlMoveFromRangeStart w:id="1"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveFromRangeEnd w:id="1"/>
  <w:moveFromRangeEnd w:id="0" w:displacedByCustomXml="prev"/>
  <w:p/>
  <w:moveToRangeStart w:id="2" w:name="move1" w:displacedByCustomXml="next"/>
  <w:customXmlMoveToRangeStart w:id="3"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveToRangeEnd w:id="3"/>
  <w:moveFromRangeEnd w:id="2" w:displacedByCustomXml="prev"/>
</w:body>
```

The customXmlMoveFromRangeStart element delimits the start of the region in which all custom XML elements have been moved from this location with revisions enabled. Since this element only affects custom XML, any text in the region is not revision marked moved by this element when present, but the corresponding physical characters for the custom XML element are. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.10 customXmlMoveToRangeEnd (Custom XML Markup Move Destination Location End)

This element specifies the end of a region within which all custom XML markup was moved to this location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move destination start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move destination applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlMoveToRangeStart element (§17.13.5.11) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant.
- If this element and its paired start occur outside of a well-formed move source container (§17.13.5.24; §17.13.5.23) with a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple end elements exist with the same id attribute value, then the document is non-conformant .

[*Example*: Consider a three-paragraph document with a single block-level custom XML markup element, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:body>
  <w:p/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:p/>
</w:body>
```

If the second paragraph is moved to the end of the document with revisions enabled. This revision must therefore be stored using the custom XML markup revision "cross structure" syntax, as follows:

```
<w:body>
  <w:p/>
  <w:moveFromRangeStart w:id="0" w:name="move1" w:displacedByCustomXml="next"
w:author="Example Author"/>
  <w:customXmlMoveFromRangeStart w:id="1"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveFromRangeEnd w:id="1"/>
  <w:moveFromRangeEnd w:id="0" w:displacedByCustomXml="prev"/>
  <w:p/>
  <w:moveToRangeStart w:id="2" w:name="move1" w:displacedByCustomXml="next"/>
  <w:customXmlMoveToRangeStart w:id="3"/>
  <w:customXml … >
    <w:p/>
  </w:customXml>
  <w:customXmlMoveToRangeEnd w:id="3"/>
  <w:moveFromRangeEnd w:id="2" w:displacedByCustomXml="prev"/>
</w:body>
```

The customXmlMoveToRangeEnd element delimits the end of the region in which all custom XML elements have been moved to this location with revisions enabled. Since this element only affects custom XML, any text in the region is not revision marked moved by this element when present, but the corresponding physical characters for the custom XML element are. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment:

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Markup) is located in §A.1. *end note*]

17.13.5.11 customXmlMoveToRangeStart (Custom XML Markup Move Destination Location Start)

This element specifies the start of a region within which all custom XML markup was moved to this location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move destination end marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move destination applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding customXmlMoveFromRangeEnd element (§17.13.5.8) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired start encapsulate a range with no custom XML markup, then the document is non-conformant.
- If this element and its paired end occur outside of a well-formed move source container (§17.13.5.24; §17.13.5.23) with a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple start elements exist with the same id attribute value, then the document is non-conformant.

[*Example*: Consider a three-paragraph document with a single block-level custom XML markup element, as follows:

```
<w:body>
  <w:p/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <w:customXml … >
      <w:p/>
    </w:customXml>
    <w:p/>
  </w:body>
```

If the second paragraph is moved to the end of the document with revisions enabled. This revision must therefore be stored using the custom XML markup revision "cross structure" syntax, as follows:

```
  <w:body>
    <w:p/>
    <w:moveFromRangeStart w:id="0" w:name="move1" w:displacedByCustomXml="next"
  w:author="Example Author"/>
    <w:customXmlMoveFromRangeStart w:id="1"/>
    <w:customXml … >
      <w:p/>
    </w:customXml>
    <w:customXmlMoveFromRangeEnd w:id="1"/>
    <w:moveFromRangeEnd w:id="0" w:displacedByCustomXml="prev"/>
    <w:p/>
    <w:moveToRangeStart w:id="2" w:name="move1" w:displacedByCustomXml="next"/>
    <w:customXmlMoveToRangeStart w:id="3"/>
    <w:customXml … >
      <w:p/>
    </w:customXml>
    <w:customXmlMoveToRangeEnd w:id="3"/>
    <w:moveFromRangeEnd w:id="2" w:displacedByCustomXml="prev"/>
  </w:body>
```

The customXmlMoveFromRangeStart element delimits the start of the region in which all custom XML elements have been moved from this location with revisions enabled. Since this element only affects custom XML, any text in the region is not revision marked moved by this element when present, but the corresponding physical characters for the custom XML element are. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</…> The author attribute specifies that the author of the current annotation is Example Author, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.12 del (Deleted Table Row)

This element specifies that the parent table row shall be treated as a deleted row whose deletion has been tracked as a revision. This setting shall not imply any revision state about the table cells in this row or their contents (which shall be revision marked independently), and shall only affect the table row itself.

[*Example*: Consider a two row by two column table in which the second row has been marked as deleted using a revision. This requirement would be specified using the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:p/>
    </w:tc>
    <w:tc>
      <w:p/>
    </w:tc>
  </w:tr>
  <w:tr>
    <w:trPr>
      <w:del w:id="0" … />
    </w:trPr>
    <w:tc>
      <w:p/>
    </w:tc>
    <w:tc>
      <w:p/>
    </w:tc>
  </w:tr>
</w:tbl>
```

The del element on the table row properties for the second table row specifies that this row was deleted, and this deletion was tracked as a revision. *end example*]

Attributes	Description
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document. If this attribute is omitted, then no author shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: `<… w:id="1" w:author="Example Author">`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	… </…> The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.13 del (Deleted Math Control Character)

This element specifies that the Office Open XML Math control character which contains this element was deleted and tracked as a revision. [*Example*: The deletion of a fraction bar. *end example*]

[*Example*: Consider a region of Office Open XML Math in a WordprocessingML document in which the control character for the fraction bar has been deleted, as follows:

$$a + \frac{b}{c}$$

This deletion is represented as the following WordprocessingML:

```
<m:f>
  <m:fPr>
    <m:ctrlPr>
      <w:del w:id="0" w:author="Joe Smith" w:date="2006-03-31T12:50:00Z">
        …
      </w:del>
    </m:ctrlPr>
  </m:fPr>
  …
</m:f>
```

The del element contains all of the content which must be treated as revision marked as deleted; in this case, the fraction bar was deleted by Joe Smith on March 31, 2006 at 12:50pm. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_MathCtrlDel) is located in §A.1. *end note*]

17.13.5.14 del (Deleted Run Content)

This element specifies that the inline-level content contained within it shall be treated as deleted content which has been tracked as a revision.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a paragraph of text in a WordprocessingML document in which one word has been deleted, as follows:

Some ~~text~~

This paragraph has the word `text` marked deleted as a revision, and is represented as the following WordprocessingML:

```
<w:p>
  <w:r>
    <w:t>Some</w:t>
  </w:r>
  <w:del w:id="0" w:author="Joe Smith" w:date="2006-03-31T12:50:00Z">
    <w:r>
      <w:delText>text</w:delText>
    </w:r>
  </w:del>
</w:p>
```

The del element contains all of the content which must be treated as revision marked as deleted; in this case, the word `text` was deleted by `Joe Smith` on March 31, 2006 at 12:50pm. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RunTrackChange) is located in §A.1. *end note*]

17.13.5.15 del (Deleted Paragraph)

This element specifies that the paragraph mark delimiting the end of a paragraph within a WordprocessingML document shall be treated as deleted (i.e. the contents of this paragraph are no longer delimited by this paragraph mark, and are combined with the following paragraph - but those contents shall not automatically be marked as deleted) as part of a tracked revision.

[*Example*: Consider a document consisting of two paragraphs (with each paragraph delimited by a pilcrow ¶), as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This·is·paragraph·one.¶

This·is·paragraph·two.¶

If the physical character delimiting the end of the first paragraph is deleted and this change is tracked as a revision, resulting in the following:

This·is·paragraph·one.This·is·paragraph·two.¶

This revision is represented using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:rPr>
      <w:del w:id="0" … />
    </w:rPr>
  </w:pPr>
  <w:r>
    <w:t>This is paragraph one.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>This is paragraph two.</w:t>
  </w:r>
</w:p>
```

The del element on the run properties for the first paragraph mark specifies that this paragraph mark was deleted, and this deletion was tracked as a revision. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.16 ins (Inserted Math Control Character)

This element specifies that the Office Open XML Math control character which contains this element was inserted and tracked as a revision. [*Example*: The insertion of a fraction bar. *end example*]

[*Example*: Consider a region of Office Open XML Math in a WordprocessingML document in which the control character for the fraction bar has been inserted, as follows:

$$a + \frac{b}{c}$$

This insertion is represented as the following WordprocessingML:

```
<m:f>
  <m:fPr>
    <m:ctrlPr>
      <w:ins w:id="0" w:author="Joe Smith" w:date="2006-03-31T12:50:00Z">
        …
      </w:ins>
    </m:ctrlPr>
  </m:fPr>
  …
</m:f>
```

The ins element contains all of the content which must be treated as revision marked as inserted; in this case, the fraction bar was inserted by `Joe Smith` on March 31, 2006 at 12:50pm. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation	Specifies the date information for an annotation within a WordprocessingML document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

Date)	The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MathCtrlIns) is located in §A.1. *end note*]

17.13.5.17 ins (Inserted Table Row)

This element specifies that the parent table row shall be treated as an inserted row whose insertion has been tracked as a revision. This setting shall not imply any revision state about the table cells in this row or their contents (which shall be revision marked independently), and shall only affect the table row itself.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a two row by two column table in which the second row has been marked as inserted using a revision. This requirement would be specified using the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      <w:p/>
    </w:tc>
    <w:tc>
      <w:p/>
    </w:tc>
  </w:tr>
  <w:tr>
    <w:trPr>
      <w:ins w:id="0" … />
    </w:trPr>
    <w:tc>
      <w:p/>
    </w:tc>
    <w:tc>
      <w:p/>
    </w:tc>
  </w:tr>
</w:tbl>
```

The ins element on the table row properties for the second table row specifies that this row was inserted, and this insertion was tracked as a revision. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.18 ins (Inserted Run Content)

This element specifies that the inline-level content contained within it shall be treated as inserted content which has been tracked as a revision.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a paragraph of text in a WordprocessingML document in which one word has been inserted, as follows:

Some text

This paragraph has the word `text` marked inserted as a revision, and is represented as the following WordprocessingML:

```
<w:p>
  <w:r>
    <w:t>Some</w:t>
  </w:r>
  <w:ins w:id="0" w:author="Joe Smith" w:date="2006-03-31T12:50:00Z">
    <w:r>
      <w:t>text</w:t>
    </w:r>
  </w:ins>
</w:p>
```

The ins element contains all of the content which must be treated as revision marked as inserted; in this case, the word `text` was inserted by `Joe Smith` on March 31, 2006 at 12:50pm. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RunTrackChange) is located in §A.1. *end note*]

17.13.5.19 ins (Inserted Numbering Properties)

This element specifies that the numbering information defined by the parent element shall be treated as numbering information which was recorded as an insertion using revisions.

[*Example*: Consider two paragraphs in a WordprocessingML document, with the words one and two respectively, as follows:

One

Two

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If numbering is then applied to these two paragraphs, and this numbering is tracked as a revision, this revision is represented using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="1" />
      <w:ins w:id="0" w:author="Joe Smith" w:date="2005-01-01T10:00:00Z" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>one</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:numPr>
      <w:ilvl w:val="0" />
      <w:numId w:val="1" />
      <w:ins w:id="0" w:author="Joe Smith" w:date="2005-01-01T10:00:00Z" />
    </w:numPr>
  </w:pPr>
  <w:r>
    <w:t>two</w:t>
  </w:r>
</w:p>
```

The ins element as a child of the numbering properties specifies that the paragraphs in this document have been given numbering properties by `Joe Smith` and that this change was marked as a revision. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Author, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.20 ins (Inserted Paragraph)

This element specifies that the paragraph mark delimiting the end of a paragraph within a WordprocessingML document shall be treated as deleted (i.e. the contents of this paragraph are no longer delimited by this paragraph mark, and are combined with the following paragraph) as part of a tracked revision.

[*Example*: Consider a document consisting of a single paragraph, as follows:

This·is·paragraph·one.This·is·paragraph·two.¶

If the first sentence is moved into its own new paragraph, and this change is tracked as a revision, resulting in the following:

This·is·paragraph·one.¶

This·is·paragraph·two.¶

This revision is represented using the following WordprocessingML:

```
<w:p>
  <w:pPr>
    <w:rPr>
      <w:ins w:id="0" … />
    </w:rPr>
  </w:pPr>
  <w:r>
    <w:t>This is paragraph one.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>This is paragraph two.</w:t>
  </w:r>
</w:p>
```

The ins element on the run properties for the first paragraph mark specifies that this paragraph mark was inserted, and this insertion was tracked as a revision. *end example*]

Attributes	Description
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document. If this attribute is omitted, then no author shall be associated with the parent annotation type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

17.13.5.21 moveFrom (Move Source Paragraph)

This element specifies that the parent paragraph has been moved away from this location and tracked as a revision. This does not imply anything about the revision state of the contents of the paragraph, and applies only to the existence of the paragraph as its own unique paragraph.

The following restrictions shall be applied to this content:

- If this element occurs outside of a move source container (§17.13.5.24; §17.13.5.23) for which a matching move destination container (§17.13.5.28; §17.13.5.27) exists in the document, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which a paragraph of text is moved down in the document. This moved paragraph would be represented using the following WordprocessingML markup:

```
<w:moveFromRangeStart w:id="0" w:name="aMove"/>
<w:p>
  <w:pPr>
    <w:rPr>
      <w:moveFrom w:id="1" … />
    </w:rPr>
  </w:pPr>
  …
</w:p>
</w:moveFromRangeEnd w:id="0"/>
```

The moveFrom element as a child of the run properties of the paragraph mark specify that this paragraph mark was part of the content which was moved in the document. This implies nothing about the contents, since they can have been added later and not tracked as a revision (they must be marked as a move using the moveFrom element (§17.13.5.22) around the run content). *end example*]

Attributes	Description
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document. If this attribute is omitted, then no author shall be associated with the parent annotation type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

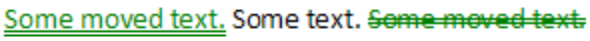
17.13.5.22 moveFrom (Move Source Run Content)

This element specifies that the inline-level content contained within it shall be treated as content which has been moved away from this location and tracked as a revision.

The following restrictions shall be applied to this content:

- If this element occurs outside of a move source container (§17.13.5.24; §17.13.5.23) for which a matching move destination container (§17.13.5.28; §17.13.5.27) exists in the document, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this image, green underline indicates the move destination and the green strikethrough indicates the move source location):



This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveFrom element specifies that all of the inline-level content contained within must be revision marked as content which was moved from its current location in the document. Because this moved content is contained within a complete move source container (moveFromRangeStart and moveFromRangeEnd) with a corresponding move destination, this content is tracked as a move. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RunTrackChange) is located in §A.1. *end note*]

17.13.5.23 moveFromRangeEnd (Move Source Location Container - End)

This element specifies the end of a region whose move source contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information shall be stored about that move source:

- A set of pieces of content which were moved - both inline-level content (§17.13.5.22) and paragraphs (§17.13.5.21)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

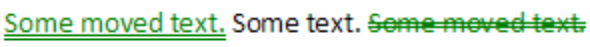
- A move source container (or "bookmark") which specifies that all content within it which marked as a move source is part of a single named move. The name attribute on the move container links a group of move source content with the corresponding group of move destination content.

This element defines the end of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding start of a move source container in the document.

The following restrictions are applied to the use of this element:

- If this element occurs without a corresponding moveFromRangeStart element (§17.13.5.24) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired end occur without a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple move source containers surround the same text, the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this image, green underline indicates the move destination and the green strikethrough indicates the move source location):



This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

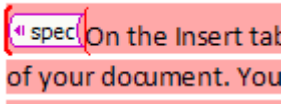
```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveFromRangeEnd element specifies the end of the move source container within which all moved content is part of the move named `move1`. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkupRange) is located in §A.1. *end note*]

17.13.5.24 moveFromRangeStart (Move Source Location Container - Start)

This element specifies the start of a region whose move source contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information shall be stored about that move source:

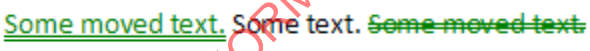
- A set of pieces of content which were moved - both inline-level content (§17.13.5.22) and paragraphs (§17.13.5.21)
- A move source container (or "bookmark") which specifies that all content within it which marked as a move source is part of a single named move. The name attribute on the move container links a group of move source content with the corresponding group of move destination content.

This element defines the start of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding end of a move source container in the document.

The following restrictions are applied to the use of this element

- If this element occurs without a corresponding moveFromRangeEnd element (§17.13.5.23) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired end occur without a matching move destination container (§17.13.5.28; §17.13.5.27), then the document is non-conformant.
- If multiple start elements exist with the same id attribute value, then the document is non-conformant.
- If multiple move source containers surround the same text, the document is non-conformant .

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this image, green underline indicates the move destination and the green strikethrough indicates the move source location):



This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveFromRangeStart element specifies the start of the move source container within which all moved content is part of the move named `move1`. *end example*]

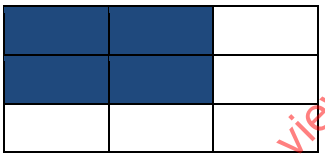
Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

colFirst (First Table Column Covered By Bookmark)	Specifies the zero-based index of the first column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	cover cells within a certain column and row range within that table, by specifying:
	• The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row.
	• The first column included in the bookmark for each of the specified row(s) via this attribute.
	• The last column included in the bookmark for each of the specified row(s) via the colLast attribute.
	• The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row.
	
	If this attribute appears, then the colLast attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored.
	
	[*Example*: Consider a three row by three column table where a table bookmark must be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`…`
	`<w:tr>`
	`<w:tc>`
	`<w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`</w:tr>`
	`<w:tr>`

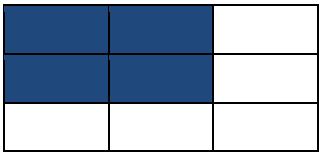
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:bookmarkEnd w:id="0" /> </w:tr> <w:tr> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colFirst attribute specifies that all columns starting with the first column must be included in the table bookmark. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
colLast (Last Table Column Covered By Bookmark)	Specifies the zero-based index of the last column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row. • The first column included in the bookmark for each of the specified row(s) via the colFirst attribute. • The last column included in the bookmark for each of the specified row(s) via this attribute. • The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	If this attribute appears, then the colFirst attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value does not equal or exceed the value of colFirst or the number of columns in the table, then both values should be ignored.
	
	[*Example*: Consider a three row by three column table where a table bookmark shall be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`…`
	`<w:tr>`
	`<w:tc>`
	`<w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`</w:tr>`
	`<w:tr>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:bookmarkEnd w:id="0" />`
	`</w:tr>`
	`<w:tr>`
	`<w:tc>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

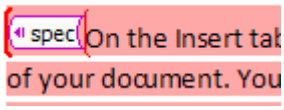
©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colLast attribute specifies that the last column that shall be included in the table bookmark is the second column. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored.
	
	[*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:
	
	 
	
	Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML:
	
	`…`
	`<w:commentRangeStart w:id="0" />`
	`<w:commentRangeStart w:id="1" w:displacedByCustomXml="next" />`
	`<w:customXml w:element="spec" … />`
	`<w:p>`
	`  …`
	`</w:p>`
	`…`
	
	The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*]
	
	The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
name (Bookmark Name)	Specifies the bookmark name. If multiple bookmarks in a document share the same name, then the first bookmark (defined by the location of the bookmarkStart element in document order) shall be maintained, and all subsequent bookmarks should be ignored. [*Example*: Consider the following XML for a bookmark around a single word: <w:p> <w:bookmarkStart w:id="0" w:name="place" /> <w:r> <w:t>Seattle</w:t> </w:r> <w:bookmarkEnd w:id="0" /> </w:p> The name attribute specifies that the name for this bookmark is place. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MoveBookmark) is located in §A.1. *end note*]

17.13.5.25 moveTo (Move Destination Run Content)

This element specifies that the inline-level content contained within it shall be treated as content which has been moved to this location and tracked as a revision.

The following restrictions shall be applied to this content:

- If this element occurs outside of a move destination container (§17.13.5.28; §17.13.5.27) for which a matching move source container (§17.13.5.24; §17.13.5.23) exists in the document, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

image, green underline indicates the move destination and the green strikethrough indicates the move source location):

Some moved text. Some text. ~~Some moved text.~~

This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveTo element specifies that all of the inline-level content contained within must be revision marked as content which was moved to its current location in the document. Because this moved content is contained within a complete move destination container (moveToRangeStart and moveToRangeEnd) with a corresponding move source, this content is tracked as a move. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</…> The author attribute specifies that the author of the current annotation is Example Author, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RunTrackChange) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.26 moveTo (Move Destination Paragraph)

This element specifies that the parent paragraph has been moved to this location and tracked as a revision. This does not imply anything about the revision state of the contents of the paragraph, and applies only to the existence of the paragraph as its own unique paragraph.

The following restrictions shall be applied to this content:

- If this element occurs outside of a move destination container (§17.13.5.28; §17.13.5.27) for which a matching move source container (§17.13.5.24; §17.13.5.23) exists in the document, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which a paragraph of text is moved down in the document. This moved paragraph would be represented using the following WordprocessingML markup:

```
<w:moveToRangeStart w:id="0" w:name="aMove"/>
<w:p>
  <w:pPr>
    <w:rPr>
      <w:moveTo w:id="1" … />
    </w:rPr>
  </w:pPr>
  …
</w:p>
</w:moveToRangeEnd w:id="0"/>
```

The moveTo element as a child of the run properties of the paragraph mark specify that this paragraph mark was part of the content which was moved in the document. This implies nothing about the contents, since they might have been added later and not tracked as a revision (they must be marked as a move using the moveTo element (§17.13.5.25) around the run content). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChange) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.13.5.27 moveToRangeEnd (Move Destination Location Container - End)

This element specifies the end of a region whose move destination contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information shall be stored about that move destination:

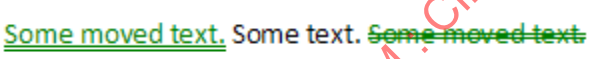
- A set of pieces of content which were moved - both inline-level content (§17.13.5.25) and paragraphs (§17.13.5.26)
- A move destination container (or "bookmark") which specifies that all content within it which marked as a move destination is part of a single named move. The name attribute on the move container links a group of move destination content with the corresponding group of move source content.

This element defines the end of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding start of a move destination container in the document.

The following restrictions are applied to the use of this element:

- If this element occurs without a corresponding moveToRangeStart element (§17.13.5.28) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired end occur without a matching move source container (§17.13.5.24; §17.13.5.23), then the document is non-conformant.
- If multiple move destination containers surround the same text, the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this image, green underline indicates the move destination and the green strikethrough indicates the move source location):



This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

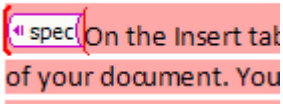
```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveToRangeEnd element specifies the end of the move destination container within which all moved content is part of the move named `move1`. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkupRange) is located in §A.1. *end note*]

17.13.5.28 moveToRangeStart (Move Destination Location Container - Start)

This element specifies the start of the region whose move destination contents are part of a single named move. When a move destination is stored as a revision in a WordprocessingML document, two pieces of information shall be stored about that move destination:

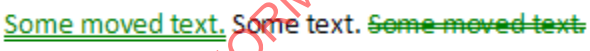
- A set of pieces of content which were moved - both inline-level content (§17.13.5.25) and paragraphs (§17.13.5.26)
- A move destination container (or "bookmark") which specifies that all content within it which marked as a move destination is part of a single named move. The name attribute on the move container links a group of move destination content with the corresponding group of move source content.

This element defines the start of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding end of a move destination container in the document.

The following restrictions are applied to the use of this element

- If this element occurs without a corresponding moveToRangeEnd element (§17.13.5.27) with a matching id attribute value, then the document is non-conformant.
- If this element and its paired end occur without a matching move source container (§17.13.5.24; §17.13.5.23), then the document is non-conformant.
- If multiple start elements exist with the same id attribute value, then the document is non-conformant.
- If multiple move destination containers surround the same text, the document is non-conformant.

[*Example*: Consider a WordprocessingML document in which the first paragraph contains two sentences, and the first sentence is moved before the second sentence, and this move is tracked as a revision, as follows (in this image, green underline indicates the move destination and the green strikethrough indicates the move source location):



This document has the sentence `Some moved text.` moved to the first sentence in the document. This revision is represented using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:moveToRangeStart w:id="0" … w:name="move1" />
  <w:moveTo w:id="1" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveTo>
  <w:moveToRangeEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve">Some text.</w:t>
  </w:r>
  <w:moveFromRangeStart w:id="2" … w:name="move1" />
  <w:moveFrom w:id="3" … >
    <w:r>
      <w:t>Some moved text.</w:t>
    </w:r>
  </w:moveFrom>
  <w:moveFromRangeEnd w:id="2" />
</w:p>
```

The moveToRangeStart element specifies the start of the move destination container within which all moved content is part of the move named `move1`. *end example*]

Attributes	Description

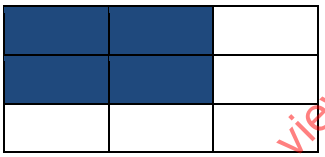
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

colFirst (First Table Column Covered By Bookmark)	Specifies the zero-based index of the first column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	cover cells within a certain column and row range within that table, by specifying:
	• The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row.
	• The first column included in the bookmark for each of the specified row(s) via this attribute.
	• The last column included in the bookmark for each of the specified row(s) via the colLast attribute.
	• The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row.
	
	If this attribute appears, then the colLast attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored.
	
	[*Example*: Consider a three row by three column table where a table bookmark must be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`…`
	`<w:tr>`
	`<w:tc>`
	`<w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`</w:tr>`
	`<w:tr>`

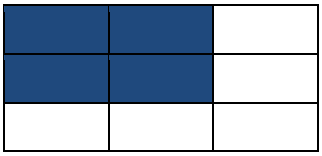
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:bookmarkEnd w:id="0" /> </w:tr> <w:tr> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colFirst attribute specifies that all columns starting with the first column must be included in the table bookmark. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
colLast (Last Table Column Covered By Bookmark)	Specifies the zero-based index of the last column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row. • The first column included in the bookmark for each of the specified row(s) via the colFirst attribute. • The last column included in the bookmark for each of the specified row(s) via this attribute. • The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

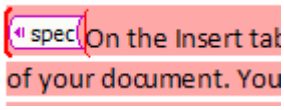
	If this attribute appears, then the colFirst attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value does not equal or exceed the value of colFirst or the number of columns in the table, then both values should be ignored.
	
	[*Example*: Consider a three row by three column table where a table bookmark shall be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`…`
	`<w:tr>`
	`<w:tc>`
	`<w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`</w:tr>`
	`<w:tr>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:bookmarkEnd w:id="0" />`
	`</w:tr>`
	`<w:tr>`
	`<w:tc>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colLast attribute specifies that the last column that shall be included in the table bookmark is the second column. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment: <… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > …

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

name (Bookmark Name)	Specifies the bookmark name.
	
	If multiple bookmarks in a document share the same name, then the first bookmark (defined by the location of the bookmarkStart element in document order) shall be maintained, and all subsequent bookmarks should be ignored.
	
	[*Example*: Consider the following XML for a bookmark around a single word:
	
	`<w:p>`
	`<w:bookmarkStart w:id="0" w:name="place" />`
	`<w:r>`
	`<w:t>Seattle</w:t>`
	`</w:r>`
	`<w:bookmarkEnd w:id="0" />`
	`</w:p>`
	
	The name attribute specifies that the name for this bookmark is `place`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MoveBookmark) is located in §A.1. *end note*]

17.13.5.29 pPrChange (Revision Information for Paragraph Properties)

This element specifies the details about a single revision to a set of paragraph properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of paragraph properties which were applied to this paragraph before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of paragraph properties).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a paragraph in a WordprocessingML document which is centered, and this change in the paragraph properties is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

```
<w:pPr>
  <w:jc w:val="center"/>
  <w:pPrChange w:id="0" w:date="01-01-2006T12:00:00" w:author="John Doe">
    <w:pPr/>
  </w:pPrChange>
</w:pPr>
```

The pPrChange element specifies that there was a revision to the paragraph properties at `01-01-2006` by `John Doe`, and the previous set of paragraph properties on the paragraph were the null set (i.e. no paragraph properties explicitly present under the pPr element). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PPrChange) is located in §A.1. *end note*]

17.13.5.30 rPrChange (Revision Information for Run Properties on the Paragraph Mark)

This element specifies the details about a single revision to a set of run properties applied to a paragraph mark within a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of run properties which were applied to this paragraph mark before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of run properties).

[*Example*: Consider an italicized paragraph mark in a WordprocessingML document which is also made bold and the latter change in the run properties is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:rPr>
  <w:b/>
  <w:i/>
  <w:rPrChange w:id="0" w:date="01-01-2006T12:00:00" w:author="John Doe">
    <w:rPr>
      <w:i/>
    </w:rPr>
  </w:rPrChange>
</w:rPr>
```

The rPrChange element specifies that there was a revision to the paragraph mark's run properties at 01-01-2006 by John Doe, and the previous set of run properties was simply the italicization using the i element (§17.3.2.16). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is Example Author, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ParaRPrChange) is located in §A.1. *end note*]

17.13.5.31 rPrChange (Revision Information for Run Properties)

This element specifies the details about a single revision to a set of run properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of run properties which were applied to this run before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of run properties).

[*Example*: Consider an italicized run in a WordprocessingML document which is also made bold and the latter change in the run properties is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:rPr>
  <w:b/>
  <w:i/>
  <w:rPrChange w:id="0" w:date="01-01-2006T12:00:00" w:author="John Doe">
    <w:rPr>
      <w:i/>
    </w:rPr>
  </w:rPrChange>
</w:rPr>
```

The rPrChange element specifies that there was a revision to the run properties at `01-01-2006` by `John Doe`, and the previous set of run properties was simply the italicization using the i element (§17.3.2.16). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`  …`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RPrChange) is located in §A.1. *end note*]

17.13.5.32 sectPrChange (Revision Information for Section Properties)

This element specifies the details about a single revision to a set of section properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of section properties which were applied to the parent section before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of section properties).

[*Example*: Consider a section in a WordprocessingML document which is set to be divided into three columns, and this change in the section properties is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:sectPr>
  <w:cols w:num="3" … >
    …
  </w:cols>
  <w:sectPrChange w:id="0" w:date="01-01-2006T12:00:00" w:author="John Doe">
    <w:sectPr/>
  </w:sectPrChange>
</w:sectPr>
```

The sectPrChange element specifies that there was a revision to the section properties at `01-01-2006` by `John Doe`, and the previous set of properties on the section were the null set (i.e. no section properties explicitly present under the sectPr element). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`  …`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`  …`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`  …`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SectPrChange) is located in §A.1. *end note*]

17.13.5.33 tblGridChange (Revision Information for Table Grid Column Definitions)

This element specifies the details about a single revision to a table's grid column definitions within a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the definition of the table grid which was applied to the parent table before this revision

[*Example*: Consider a two column table in a WordprocessingML document which has the width of its first column significantly reduced, and this change in the table grid is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:tblGrid>
  <w:gridCol w:w="1548" />
  <w:gridCol w:w="8028" />
  <w:tblGridChange w:id="1">
    <w:tblGrid>
      <w:gridCol w:w="4788" />
      <w:gridCol w:w="4788" />
    </w:tblGrid>
  </w:tblGridChange>
</w:tblGrid>
```

The tblGridChange element specifies that there was a revision to the table grid, and the previous table grid had both columns with a width of 4788 twentieths of a point, vs. their current widths of 1548 and 8028 twentieths of a point respectively. *end example*]

Attributes	Description

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblGridChange) is located in §A.1. *end note*]

17.13.5.34 tblPrChange (Revision Information for Table Properties)

This element specifies the details about a single revision to a set of table properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table properties which were applied to the parent table before this revision

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table properties).

[*Example*: Consider a table in a WordprocessingML document which has the associated table style changed from `LightList` to `LightShading`, and this change in the table properties is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

```
<w:tblPr>
  <w:tblStyle w:val="LightShading"/>
  <w:tblW w:w="0" w:type="auto"/>
  <w:tblLook w:firstRow="true" w:firstColumn="true"
      w:noVBand="true" />
  <w:tblPrChange w:id="0" w:author="Tristan Davis" w:date="2006-06-
01T13:39:00Z">
    <w:tblPr>
      <w:tblStyle w:val="LightList"/>
      <w:tblW w:w="0" w:type="auto"/>
      <w:tblLook w:firstRow="true" w:firstColumn="true"
      w:noVBand="true" />
    </w:tblPr>
  </w:tblPrChange>
</w:tblPr>
```

The tblPrChange element specifies that there was a revision to the table properties at `2006-06-01` by `Tristan Davis`, and the previous set of properties on the table was the set specifies in the child tblPr element (including the table style of `LightList`). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" …>`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrChange) is located in §A.1. *end note*]

17.13.5.35 tblPrExChange (Revision Information for Table-Level Property Exceptions)

This element specifies the details about a single revision to a set of table-level property exceptions in a WordprocessingML document.

This element stores this revision as follows:

©ISO/IEC 2016 – All rights reserved

- The child element of this element contains the complete set of table-level property exceptions which were applied to the parent table row before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table-level exception properties).

[*Example*: Consider a set of table rows which are part of a table in a WordprocessingML document, have table-level property exceptions, and this change in the table-level properties to a fixed table width of ten inches is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

```
<w:tblPrEx>
  <w:tblW w:w="14400" w:type="dxa"/>
  <w:tblPrExChange w:id="0" w:author="Tristan Davis" w:date="2006-06-
01T13:39:00Z">
    <w:tblPrEx>
      <w:tblW w:w="0" w:type="auto"/>
    </w:tblPrEx>
  </w:tblPrExChange>
</w:tblPrEx>
```

The tblPrExChange element specifies that there was a revision to the table-level property exceptions at `2006-06-01` by `Tristan Davis`, and the previous set of table-level property exceptions set specifies in the child tblPrEx element. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`
	
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TblPrExChange) is located in §A.1. *end note*]

17.13.5.36 tcPrChange (Revision Information for Table Cell Properties)

This element specifies the details about a single revision to a set of table cell properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table cell properties which were applied to the parent table before this revision

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table cell properties).

[*Example*: Consider a table cell in a WordprocessingML document which has a change in the table cell properties that is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

```
<w:tcPr>
  <w:cnfStyle w:firstColumn="true" />
  <w:tcW w:w="3192" w:type="dxa"/>
  <w:tcPrChange w:id="8" w:author="Tristan Davis" w:date="2006-06-01T13:39:00Z">
    <w:tcPr>
      <w:tcW w:w="3192" w:type="dxa"/>
    </w:tcPr>
  </w:tcPrChange>
</w:tcPr>
```

The tcPrChange element specifies that there was a revision to the table cell properties at `2006-06-01` by `Tristan Davis`, and the previous set of properties on the table cell was the set specifies in the child tcPr element. *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500. If this attribute is omitted, then no date information shall be associated with the parent annotation type. [*Example*: Consider a comment represented using the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… w:id="1" w:date="2006-01-01T10:00:00"> … </…> The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TcPrChange) is located in §A.1. *end note*]

17.13.5.37 trPrChange (Revision Information for Table Row Properties)

This element specifies the details about a single revision to a set of table row properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table row properties which were applied to the parent table row before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table row properties).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a table cell in a WordprocessingML document which has a change in the table row properties that is tracked as a revision. This revision would be specified using the following WordprocessingML markup:

```
<w:trPr>
  <w:cantSplit/>
  <w:trPrChange w:id="8" w:author="Tristan Davis" w:date="2006-06-01T13:39:00Z">
    <w:trPr/>
  </w:trPrChange>
</w:trPr>
```

The trPrChange element specifies that there was a revision to the table row properties at `2006-06-01` by `Tristan Davis`, and the previous set of properties on the table row was the set specified in the child trPr element (in this case, the null set). *end example*]

Attributes	Description

author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document.
	
	If this attribute is omitted, then no author shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:author="Example Author">`
	`…`
	`</…>`
	
	The author attribute specifies that the author of the current annotation is `Example Author`, which can be used as desired. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of ISO/IEC 29500.
	
	If this attribute is omitted, then no date information shall be associated with the parent annotation type.
	
	[*Example*: Consider a comment represented using the following WordprocessingML fragment:
	
	`<… w:id="1" w:date="2006-01-01T10:00:00">`
	`…`
	`</…>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The date attribute specifies that the date of the current annotation is January 1st 2006 at 10:00 AM, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_DateTime simple type (§17.18.9).

id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`…`
	`</…>`
	
	The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrPrChange) is located in §A.1. *end note*]

17.13.6 Bookmarks

Within a WordprocessingML document, *bookmarks* refer to arbitrary regions of content which are bounded and have a unique name associated with them.

Because bookmarks are a legacy word processing function which predates the concepts of XML and well-formedness, they can start and end at any location within a document's contents and therefore shall use the "cross-structure" annotation format described in §17.13.2.

[*Example*: Consider the following WordprocessingML markup for two paragraphs, each reading `Example Text`, where a bookmark has been added spanning the second word in paragraph one and the first word in paragraph two:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t>Example</w:t>
  </w:r>
  <w:bookmarkStart w:id="0" w:name="sampleBookmark" />
  <w:r>
    <w:t xml:space="preserve"> text.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t>Example</w:t>
  </w:r>
  <w:bookmarkEnd w:id="0" />
  <w:r>
    <w:t xml:space="preserve"> text.</w:t>
  </w:r>
</w:p>
```

The bookmarkStart and bookmarkEnd elements (§17.13.6.2; §17.13.6.1) specify the location where the bookmark starts and ends, but cannot contain it using a single tag because it spans part of two paragraphs. However, the two tags are part of one group because the id attribute value specifies 0 for both. *end example*]

17.13.6.1 bookmarkEnd (Bookmark End)

This element specifies the end of a bookmark within a WordprocessingML document. This end marker is matched with the appropriately paired start marker by matching the value of the id attribute from the associated bookmarkStart element.

If no bookmarkStart element exists prior to this element in document order with a matching id attribute value, then the document is non-conformant.

[*Example*: Consider a document with a bookmark which spans half of paragraph one, and part of paragraph two. The following WordprocessingML illustrates an example of content which fufills this constraint:

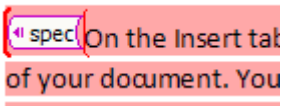
```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is sentence one.</w:t>
  </w:r>
  <w:bookmarkStart w:id="0" w:name="testing123"/>
  <w:r>
    <w:t>This is sentence two.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This </w:t>
  </w:r>
  <w:bookmarkEnd w:id="0"/>
  <w:r>
    <w:t>is sentence three.</w:t>
  </w:r>
</w:p>
```

The bookmarkEnd element specifies the end of the region for the bookmark whose bookmarkStart element has an id attribute value of `0`. In this case, this refers to the `testing123` bookmark. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … `<w:commentRangeStart w:id="0" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkupRange) is located in §A.1. *end note*]

17.13.6.2 bookmarkStart (Bookmark Start)

This element specifies the start of a bookmark within a WordprocessingML document. This start marker is matched with the appropriately paired end marker by matching the value of the id attribute from the associated bookmarkEnd element.

If no bookmarkEnd element exists subsequent to this element in document order with a matching id attribute value, then the document is non-conformant.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If a bookmark begins and ends within a single table, it is possible for that bookmark to cover discontiguous parts of that table which are logically related (e.g. a single column in a table). This kind of placement for a bookmark is accomplished (and described in detail) on the colFirst and colLast attributes on this element.

[*Example*: Consider a document with a bookmark which spans half of paragraph one, and part of paragraph two. The following WordprocessingML illustrates an example of content which fufills this constraint:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is sentence one.</w:t>
  </w:r>
  <w:bookmarkStart w:id="0" w:name="testing123"/>
  <w:r>
    <w:t>This is sentence two.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t xml:space="preserve">This </w:t>
  </w:r>
  <w:bookmarkEnd w:id="0"/>
  <w:r>
    <w:t>is sentence three.</w:t>
  </w:r>
</w:p>
```

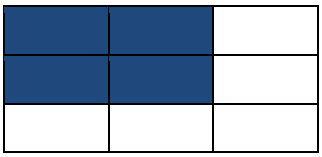
The bookmarkStart element specifies the start of the region for the `testing123` bookmark. This element is then linked to the bookmarkEnd element which also has an id attribute value of `0`. *end example*]

Attributes	Description
colFirst (First Table Column Covered By Bookmark)	Specifies the zero-based index of the first column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row. • The first column included in the bookmark for each of the specified row(s) via this attribute. • The last column included in the bookmark for each of the specified row(s) via the colLast attribute. • The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute appears, then the colLast attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this attribute and its pair are not present on a bookmark which is contained in a table, the bookmark shall apply to the entire table.
	
	If this element occurs on a bookmark which is contained in a table, but this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored and the bookmark shall apply to the entire table.
	
	[*Example*: Consider a three row by three column table where a table bookmark must be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`…`
	`<w:tr>`
	`<w:tc>`
	`<w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`</w:tr>`
	`<w:tr>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`
	`</w:tc>`
	`<w:tc>`
	`<w:p/>`

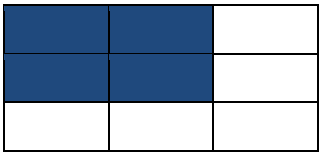
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</w:tc> <w:bookmarkEnd w:id="0" /> </w:tr> <w:tr> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colFirst attribute specifies that all columns starting with the first column must be included in the table bookmark. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
colLast (Last Table Column Covered By Bookmark)	Specifies the zero-based index of the last column in this row which shall be part of this bookmark. When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row. • The first column included in the bookmark for each of the specified row(s) via the colFirst attribute. • The last column included in the bookmark for each of the specified row(s) via this attribute. • The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row. If this attribute appears, then the colFirst attribute shall also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this attribute and its pair are not present on a bookmark which is contained in a table, the bookmark shall apply to the entire table. If this element occurs on a bookmark which is contained in a table, but this value does not equal or exceed the value of colFirst or the number of columns in the table, then

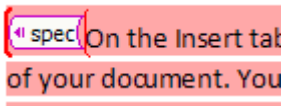
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	both values should be ignored and the bookmark shall apply to the entire table.
	
	[*Example*: Consider a three row by three column table where a table bookmark shall be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):
	
	 
	
	This bookmark would be specified using the following WordprocessingML for the table's conents:
	
	`<w:tbl>`
	`  …`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:bookMarkStart w:colFirst="0" w:colLast="1" w:id="0" w:name="table"/>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:bookmarkEnd w:id="0" />`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</w:tc> <w:tc> <w:p/> </w:tc> </w:tr> </w:tbl> The colLast attribute specifies that the last column that shall be included in the table bookmark is the second column. This applies starting with the first row and ending with the second row (the two rows within the bookmark's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" />

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).
id (Annotation Identifier)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
name (Bookmark Name)	Specifies the bookmark name. If multiple bookmarks in a document share the same name, then the first bookmark (defined by the location of the bookmarkStart element in document order) shall be maintained, and all subsequent bookmarks should be ignored. [*Example*: Consider the following XML for a bookmark around a single word: <w:p> <w:bookmarkStart w:id="0" w:name="place" /> <w:r> <w:t>Seattle</w:t> </w:r> <w:bookmarkEnd w:id="0" /> </w:p>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The name attribute specifies that the name for this bookmark is `place`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bookmark) is located in §A.1. *end note*]

17.13.7 Range Permissions

Range permissions in a WordprocessingML document refer to a special kind of bookmark used to control which subset(s) of users can edit a particular region of a document. Range permissions specify the user or set of users which are allowed to edit all content between them whenever the document protection specified by the documentProtection element (§17.15.1.29) is enabled and set to `readOnly` or `comments`.

Like bookmarks, range permissions are a legacy word processing function which predates the concepts of XML and well-formedness, so they can start and end at any location within a document's contents and therefore shall use the "cross-structure" annotation format described in §17.13.2.

[*Example*: Consider the following WordprocessingML markup for a single paragraph, where a range permission has been added spanning the words `range permission`:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is a </w:t>
  </w:r>
  <w:permStart w:id="0" w:edGrp="everyone"/>
  <w:r>
    <w:t>range permission</w:t>
  </w:r>
  <w:permEnd w:id="0"/>
  <w:r>
    <w:t>.</w:t>
  </w:r>
</w:p>
```

The permStart and permEnd elements (§17.13.7.1; §17.13.7.2) specify the location where the range permission starts and ends. The two tags are part of one group because the id attribute value specifies `0` for both.

If document protection was enabled, then no content in this document must be editable except for this range permission, which is editable by all users that open the document (specified using an editor group of everyone). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.13.7.1 permEnd (Range Permission End)

This element specifies the end of a single range permission within a WordprocessingML document. This end marker is matched with the appropriately paired start marker by matching the value of the id attribute from the associated permStart element.

If no permStart element exists prior to this element in document order with a matching id attribute value, then the document is non-conformant.

[*Example*: Consider a document with a range permission which spans half of paragraph one, and part of paragraph two. The following WordprocessingML illustrates an example of content which fufills this constraint:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is sentence one.</w:t>
  </w:r>
  <w:permStart w:id="0" w:edGrp="everyone"/>
  <w:r>
    <w:t>This is sentence two.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:r>
    <w:t xml:space="preserve">This </w:t>
  </w:r>
  <w:permEnd w:id="0"/>
  <w:r>
    <w:t>is sentence three.</w:t>
  </w:r>
</w:p>
```

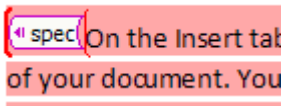
The permEnd element specifies the end of the region for the range permission whose permStart element has an id attribute value of 0. *end example*]

Attributes	Description
displacedByCustomXml (Annotation Displaced By Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored.
	
	[*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:
	
	 
	
	Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML:
	
	`…`
	`<w:commentRangeStart w:id="0" />`
	`<w:commentRangeStart w:id="1" w:displacedByCustomXml="next" />`
	`<w:customXml w:element="spec" … />`
	`<w:p>`
	`  …`
	`</w:p>`
	`…`
	
	The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*]
	
	The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).

id (Annotation ID)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element.
	
	If this attribute is omitted, then the document is non-conformant.
	
	[*Example*: Consider an annotation represented using the following WordprocessingML fragment:
	
	`<… w:id="1" … >`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Perm) is located in §A.1. *end note*]

17.13.7.2 permStart (Range Permission Start)

This element specifies the start of a range permission within a WordprocessingML document. This start marker is matched with the appropriately paired end marker by matching the value of the id attribute from the associated permEnd element.

If no permEnd element exists subsequent to this element in document order with a matching id attribute value, then the document is non-conformant.

If a range permission begins and ends within a single table, it is possible for that permission to cover discontiguous parts of that table which are logically related (e.g. a single column in a table). This type of placement for a range permission is accomplished (and described in detail) on the colFirst and colLast attributes on this element.

[*Example*: Consider a document with a range permission which spans half of paragraph one, and part of paragraph two. The following WordprocessingML illustrates an example of content which fufills this constraint:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This is sentence one.</w:t>
  </w:r>
  <w:permStart w:id="0" w:edGrp="everyone"/>
  <w:r>
    <w:t>This is sentence two.</w:t>
  </w:r>
</w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

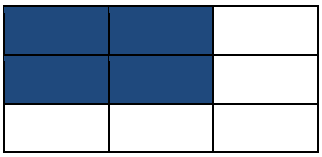
```
<w:p>
  <w:r>
    <w:t xml:space="preserve">This </w:t>
  </w:r>
  <w:permEnd w:id="0"/>
  <w:r>
    <w:t>is sentence three.</w:t>
  </w:r>
</w:p>
```

The permStart element specifies the start of the region for the range permission. This element is then linked to the permEnd element which also has an id attribute value of `0`. *end example*]

Attributes	Description
colFirst (First Table Column Covered By Range Permission)	Specifies the zero-based index of the first column in this row which shall be part of this range permission. When a range permission is contained within a table, it is possible for that range permission to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table range permission. This is accomplished by placing the permStart element in the first table cell in that row. • The first column included in the range permission for each of the specified row(s) via this attribute. • The last column included in the range permission for each of the specified row(s) via the colLast attribute. • The last row for which the specified columns are part of the table range permission. This is accomplished by placing the permEnd element at the end of that table row. If this attribute appears, then the colLast attribute shall also appear (regardless of where this range permission is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a range permission which is not contained in a table, then their values should be ignored. If this attribute and its pair are not present on a range permission which is contained in a table, the range permission shall apply to the entire table. If this element occurs on a range permission which is contained in a table, but this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored and the range permission shall apply to the entire table. [*Example*: Consider a three row by three column table where a table range permission shall be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below):

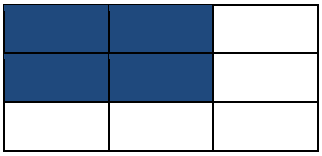
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	
	
	This range permission would be specified using the following WordprocessingML for the table's contents:
	
	`<w:tbl>`
	`  …`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:permStart w:colFirst="0" w:colLast="1" w:id="0" w:edGrp="everyone"/>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:permEnd w:id="0" />`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  </w:tr>`
	`</w:tbl>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The colFirst attribute specifies that all columns starting with the first column must be included in the table range permission. This applies starting with the first row and ending with the second row (the two rows within the range permission's start and end). *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
colLast (Last Table Column Covered By Range Permission)	Specifies the zero-based index of the last column in this row which shall be part of this range permission. When a range permission is contained within a table, it is possible for that range permission to only cover cells within a certain column and row range within that table, by specifying: • The first row for which the specified columns are part of the table range permission. This is accomplished by placing the permStart element in the first table cell in that row. • The first column included in the range permission for each of the specified row(s) via the colFirst attribute. • The last column included in the range permission for each of the specified row(s) via this attribute. • The last row for which the specified columns are part of the table range permission. This is accomplished by placing the permEnd element at the end of that table row. If this attribute appears, then the colFirst attribute shall also appear (regardless of where this range permission is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a range permission which is not contained in a table, then their values should be ignored. If this attribute and its pair are not present on a range permission which is contained in a table, the range permission shall apply to the entire table. If this element occurs on a range permission which is contained in a table, but this value does not equal or exceed the value of colFirst or the number of columns in the table, then both values should be ignored and the range permission shall apply to the entire table. [*Example*: Consider a three row by three column table where a table range permission must be applied to the contents of the first two cells in the first two rows in the table (the cells shaded below): 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

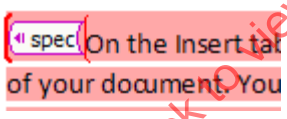
©ISO/IEC 2016 – All rights reserved

Attributes	Description

	This range permission would be specified using the following WordprocessingML for the table's contents:
	
	`<w:tbl>`
	`  …`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:permStart w:colFirst="0" w:colLast="1" w:id="0" w:edGrp="everyone"/>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:permEnd w:id="0" />`
	`  </w:tr>`
	`  <w:tr>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  <w:tc>`
	`  <w:p/>`
	`  </w:tc>`
	`  </w:tr>`
	`</w:tbl>`
	
	The colLast attribute specifies that the last column that must be included in the table range permission is the second column. This applies starting with the first row and ending with the second row (the two rows within the range permission's start and end). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
displacedByCustomXml (Annotation Displaced By Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element can differ. Specifically, in this case, the custom XML is presented *around* the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this attribute is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this attribute is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. [*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:  Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML: … <w:commentRangeStart w:id="0" /> <w:commentRangeStart w:id="1" w:displacedByCustomXml="next" /> <w:customXml w:element="spec" … /> <w:p> … </w:p> … The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	element (the spec element). *end example*] The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§17.18.13).

ed (Single User For Range Permission)	Specifies a single user for which this range permission shall be enabled (i.e. a user which shall be able to edit this range when document protection is enabled).
	
	This editor can be stored in one of the following forms:
	• DOMAIN\username - for users whose access shall be authenticated using the current user's domain credentials
	• user@domain.com - for users whose access shall be authenticated using the user's e-mail address as credentials
	• user - for users whose access shall be authenticated using the current user's machine credentials
	
	If neither the ed nor edGrp attributes are present, range permission shall not be enabled on this range.
	
	[*Example*: Consider a range permission defined as follows:
	
	`<w:permStart w:id="0" w:ed="example@example.com" … />`
	`…`
	`<w:permEnd w:id="0" />`
	
	The ed attribute value of example@example.com specifies that only user(s) who can authenticate with an application as associated with that e-mail address must be allowed to edit the contents between the start and end markers when document protection is being enforced. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

edGrp (Editor Group For Range Permission)	Specifies an alias (or editing group) which shall be used to determine if the current user shall be allowed to edit this range of the document. This mechanism simply provides a set of predefined editing groups which can be associated with user accounts by applications in any desired manner.
	
	If both the edGrp and ed attributes are present, the value of edGrp is ignored.
	
	[*Example*: Consider a range permission defined as follows:
	
	`<w:permStart w:id="0" w:edGrp="editors" … />`
	`…`
	`<w:permEnd w:id="0" />`
	
	The edGrp attribute value of editors specifies that only user(s) who the current

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	application associates with the editors group shall be allowed to edit the contents between the start and end markers when document protection is being enforced. *end example*] The possible values for this attribute are defined by the ST_EdGrp simple type (§17.18.21).
id (Annotation ID)	Specifies a unique identifier for an annotation within a WordprocessingML document. The restrictions on the id attribute, if any, are defined by the parent XML element. If this attribute is omitted, then the document is non-conformant. [*Example*: Consider an annotation represented using the following WordprocessingML fragment: <… w:id="1" … > … </…> The id attribute specifies that the ID of the current annotation is 1. This value is used to uniquely identify this annotation within the document content. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PermStart) is located in §A.1. *end note*]

17.13.8 Spelling and Grammar

The remaining kind of annotation stored in a WordprocessingML document, *spelling and grammar errors* are annotations used to specify the locations of existing spelling and grammatical errors within the contents of a document.

[*Rationale*: When a WordprocessingML document is saved, applications can choose to save currently flagged spelling and grammar errors, for two reasons:

- In order to increase the performance subsequent loads of the document (as those load operations can rely on the persisted proofing state of the document)
- In order to store words which must not be marked as proofing errors regardless of how they would normally be flagged by the proofing tools engine (i.e. to store spelling and grammar exceptions).

end rationale]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following paragraph consisting of two misspelled words, where the second word has been explicitly flagged as not being a spelling error. This paragraph would consist of the following WordprocessingML markup:

```
<w:p>
  <w:proofErr w:type="spellStart"/>
  <w:r>
    <w:t>erqwt</w:t>
  </w:r>
  <w:proofErr w:type="spellEnd"/>
  <w:r>
    <w:t xml:space="preserve"> werewr</w:t>
  </w:r>
</w:p>
```

The proofErr elements, with a val attribute value of `spellStart` and `spellEnd` respectively, delimit the start and end the content in this paragraph which is stored as a spelling error. Since the second word is not included in that range, it is not stored as a spelling error. *end example*]

17.13.8.1 proofErr (Proofing Error Anchor)

This element specifies the presence of a start or end anchor for a single proofing error within a WordprocessingML document.

When proofing errors are stored in a document, their semantics shall be interpreted as follows:

- Each proofing error with a type attribute value of `spellStart` shall be linked with the next error with a type attribute of `spellEnd`. If one does not exist, then this error should be ignored.
- Each proofing error with a type attribute value of `spellEnd` which was not preceded by an error with a type attribute value of `spellStart` (that was not previously matched to an end) should be ignored.
- Each proofing error with a type attribute value of `gramStart` shall be linked with the next error with a type attribute of `gramEnd`. If one does not exist, then this error should be ignored.
- Each proofing error with a type attribute value of `gramEnd` which was not preceded by an error with a type attribute value of `gramStart` (that was not previously matched to an end) should be ignored.

[*Example*: Consider the following sentence with a grammatical error in its subject/verb agreement. If an application recognized this error and wished to persist it to the document, this paragraph would consist of the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:proofErr w:type="gramStart"/>
  <w:r>
    <w:t>This are</w:t>
  </w:r>
  <w:proofErr w:type="gramEnd"/>
  <w:r>
    <w:t xml:space="preserve"> an error.</w:t>
  </w:r>
</w:p>
```

The proofErr elements, with a val attribute value of gramStart and gramEnd respectively, delimit the start and end the content in this paragraph which is stored as a grammatical error. *end example*]

Attributes	Description

type (Proofing Error Anchor Type)	Specifies the type of proofing error anchor at this location in the document. This proofing error type implies the necessary semantics for this element as defined by the parent element.
	
	[*Example*: Consider the following sentence with a proofing error, consisting of the following WordprocessingML markup:
	
	`<w:r>`
	`  <w:t>are</w:t>`
	`</w:r>`
	`<w:proofErr w:val="gramEnd"/>`
	`<w:r>`
	`  <w:t xml:space="preserve"> an error.</w:t>`
	`</w:r>`
	
	The val attribute value of gramEnd specifies that the proofing error is the location of the end of content which is stored as a grammatical error. *end example*]
	
	The possible values for this attribute are defined by the ST_ProofErr simple type (§17.18.70).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ProofErr) is located in §A.1. *end note*]

17.14 Mail Merge

Mail merge refers to an operation by which WordprocessingML documents can work in conjunction with data from an external data source, importing this data into a document according to a set of codes contained in WordprocessingML known as fields.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

A WordprocessingML document that contains the mailMerge element (§17.14.20) and is therefore connected to an external data source, is known as a *source document*. In addition to being connected to an external data source and containing fields, a source document can contain any regular WordprocessingML constructs such as:

- Text runs
- Paragraphs
- Images
- Tables
- Numbering
- Etc.

There are two key parts of the mail merge data stored in a WordprocessingML document

9. Information connecting a document to an external data source
10. Information populating fields within that document with external data.

Once the fields in a merged document have been populated with external data, mail merge has been completed and the resulting files are known as *mail merged documents* or simply *merged documents*.

The mail merge settings for a WordprocessingML document are stored in two locations:

- The standard mail merge settings are stored as the child elements of the mailMerge element (§17.14.20)
- A set of additional mail merge settings stored in the odso element (§17.14.25), and collectively referred to as the Office Data Source Object settings. The *Office Data Source Object* is an extension to the standard settings stored with a mail merge which performs two functions: First, it provides additional information about the mail merge data source, specifically: information about how to map the columns in the data source to `MERGEFIELD` fields and information about records which shall be included and excluded when creating merged documents. Second, it provides an alternate set of connection information which should be used when the dataType element (§17.14.10) specifies a value of `native`. This alternate connection string provides additional connection information for applications which choose to support the ODSO connection string syntax.

[*Example*: Consider a WordprocessingML document containing static WordprocessingML constructs such as text runs and paragraphs in addition to two WordprocessingML `MERGEFIELD` fields (§17.16.5.35) calling for `Courtesy Title` and `Last Name` data. The field codes for each field are displayed, delimited by {} characters:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Dear {MERGEFIELD "Courtesy Title" \m} {MERGEFIELD "Last Name" \m},

Sample text. Sample text.

Sincerely,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If the following WordprocessingML was added to this document, this document would become a *source document* rather than just a standard WordprocessingML document, as the mailMerge (§17.14.20) element specifies the elements and attributes necessary to enabled the document to connect to an external spreadsheet data source.

```
<w:mailMerge>
  …
  <w:dataType w:val="spreadsheet" />
  <w:query w:val="SELECT * FROM `Sheet1$`" />
  <w:dataSource r:id="rId1" />
  …
</w:mailMerge>
```

Here, the dataType (§17.14.10) and dataSource (§17.14.9) elements specify that the given document must be connected to the external data source target by the relationship whose relationship value is `rId1` as specified in the dataSource element (§17.14.9). While connected to the external data source, the *source document* together with the hosting application and/or data source access application extracts data from the external data source to perform the merge as specified by the connectString (§17.14.8) and query (§17.14.26) elements. *end example*]

17.14.1 active (Record Is Included in Mail Merge)

This element specifies whether a specific record from the specified external data source shall be imported into a merged WordprocessingML document when the mail merge defined for a source document is performed. If this

©ISO/IEC 2016 – All rights reserved

element's val attribute is false, then the record specified by the parent element shall not used to create a merged document.

If this element is omitted for a given record, the data record associated with it shall be imported into a merged WordprocessingML document when the mail merge is performed.

[*Example*: Consider the following fragment from a source WordprocessingML document that is connected to an external data source containing two records, one of which is not imported into a merged WordprocessingML document when the conforming hosting application performs the data import.

```
<w:recipients>
  <w:recipientData>
    <w:active w:val="false" />
    …
    <w:uniqueTag>1126664175</w:uniqueTag>
  </w:recipientData>
  <w:recipientData>
    …
    <w:uniqueTag>1530576378</w:uniqueTag>
  </w:recipientData>
</w:recipients>
```

In this XML fragment, the external data record who is identified by the uniqueTag element (§17.14.35) with a val attribute equal to 1126664175 is not imported into a merged document as the active element associated with it has a val attribute equal to false. Conversely, the external data record associated with the uniqueTag element with a val attribute equal to 1530576378 is imported into a merged document, as its active element has been omitted (implying the default value of true). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.14.2 activeRecord (Record Currently Displayed In Merged Document)

This element specifies that the hosting application shall display the given record from the specified external data source in place of the MERGEFIELD fields (§17.16.5.35) its data is mapped to via the fieldMapData element (§17.14.15) in a merged document. When this element is present, the val attribute shall specify the one-based index of the record from that data source which shall be used to populate this document.

If the activeRecord element is omitted with the viewMergedData element's val attribute equal to true, the hosting application shall behave as if the activeRecord element's val attribute was equal to 1. If the viewMergedData element (§17.14.36) is omitted or present with a val attribute equal to 0 or false, then this element shall be ignored. If the activeRecord record is given a val attribute that is less than one or greater than the number of records in the specified external data source, the hosting application shall treat this val attribute as if it were equal to 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a merged WordprocessingML document containing two WordprocessingML fields calling for `Courtesy Title` and `Last Name` data and a sample text paragraph. Also, note that the external data source this merged document is connected to contains two records, both containing name and address information, with the first record pertaining to `Mr. Doe`, and the second pertaining to `Ms. Smith`.

This table below illustrates the necessary WordprocessingML to display applicable data from the specified external data source within the merged document where fields have been inserted:

`<w:viewMergedData val="0"/>`	`<w:viewMergedData val="1" />` `<w:activeRecord w:val="1" />`	`<w:viewMergedData val="1" />` `<w:activeRecord w:val="2" />`
Dear {MERGEFIELD "Courtesy Title" \m} {MERGEFIELD "Last Name" \m}, Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,	Dear Mr. Doe: Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,	Dear Ms. Smith: Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,

end example]

[*Example*: Consider the following WordprocessingML from a merged WordprocessingML document:

```
<w:viewMergedData />
<w:activeRecord w:val="2" />
```

The activeRecord element is present and has a val attribute equal to 2, therefore this WordprocessingML specifies that a conforming hosting application must display data from the second record of the specified external data source in place of fields its data is mapped to within the merged document. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.14.3 addressFieldName (Column Containing E-mail Address)

This element specifies the column within a given external data source that contains e-mail addresses. This element is specified independently of the field mappings specified for a given merged document via the fieldMapData element (§17.14.15).

If this element is omitted, or no column exists in the data source with this column name, then the source document specifies that no e-mail address data shall be associated with this mail merge.

[*Note*: This element is generally used to allow the e-mailing of merged documents resulting from populating the fields within a merged document with external data.

This element is independent of the field mapping specified for a given merged document via the fieldMapData element (§17.14.15). This separation enables applications to email the documents resulting from the population of WordprocessingML fields with external data regardless of the presence or absence of a field mapped to external data specifying email addresses. *end note*]

[*Example*: Consider a merged WordprocessingML document that is connected to an external data source containing a column of data tilted `Alternate Email Addresses`. The following WordprocessingML would be included in the source and merged documents to specify which column in the external data source contains email addresses.

```
<w:addressFieldName w:val="Alternate Email Address" />
```

The addressFieldName element specifies that the `Alternate Email Address` column contains e-mail addresses for each record. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.4 checkErrors (Mail Merge Error Reporting Setting)

This element specifies the type of error reporting which shall be conducted by an application when performing a mail merge against the specified source data.

The type of error reporting implied by this element shall be defined as follows:

- Simulate the population of fields with mapped external data and report errors in a new document if the val attribute is equal to 1.
- While populating fields with mapped external data, pausing to report each error as it occurs if the val attribute is equal to 2.
- Populate fields with mapped external data and report errors in a new document if the val attribute is equal to 3.
- Application-defined behaviors can be used if the val attribute is equal to any other value.

If this element is omitted, or its value is set to a value outside of those specified below that is not understood by the hosting application, then its value shall be assumed to be 2.

[*Example*: Consider a mail merge whose WordprocessingML definition includes the following:

```
<w:checkErrors w:val="3" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The presence of a checkErrors element with a val attribute of 3 indicates that the hosting application must conduct the type of error reporting specified above, performing the mail merge, populating fields with mapped external data and reporting errors in a new document . *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.14.5 colDelim (Column Delimiter for Data Source)

This element specifies the character which shall be interpreted as the column delimiter used to separate columns within external data sources. The character representing the specific delimiter used for the external data source referenced by a source or merged WordprocessingML document is specified via a decimal number representing the decimal number for the Unicode character representation within this element's val attribute.

If this element is omitted, then no column delimiter shall be specified for the data source in this mail merge.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:colDelim w:val="44" />
```

Here, the colDelim element's val attribute specifies that the given external data source is using the comma character (,) to delimit column data, as 44 is the decimal value for the Unicode character representation of a comma. *end Example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.14.6 column (Index of Column Being Mapped)

This element specifies the zero-based index of the column within a given external data source which shall be mapped to the local name of a specific `MERGEFIELD` field (§17.16.5.35) specified by the parent field mapping data. The val attribute specifies this index value, which is used to look up the appropriate column in the data source.

If this element is omitted, or its value exceeds the number of columns in the associated data source, then the index of the referenced column shall be assumed to be `0`.

[*Example*: Consider a source document that is connected to an external data source with three columns. Within this external data source, these are three columns are ordered and titled as follows: `first`, `middle`, and `last`, respectively. The following WordprocessingML specifies that when this document was connected to the data source, these columns were ordered in this manner:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:fieldMapData>
  …
  <w:name w:val="first" />
  <w:column w:val="0" />
</w:fieldMapData>
<w:fieldMapData>
  …
  <w:name w:val="middle" />
  <w:column w:val="1" />
</w:fieldMapData>
<w:fieldMapData>
  …
  <w:name w:val="last" />
  <w:column w:val="2" />
</w:fieldMapData>
```

The WordprocessingML above demonstrates that the column titled `first` must be associated with the first column in the external database by specifying a column element with its val attribute equal to `0`. In addition, the column titled `middle` must be associated with the second column in the external database by specifying a column element with its val attribute equal to `1`. Finally, the column titled `last` must be associated with the third column in the external database by specifying a column element with its val attribute equal to `2`. *end example*]

It is important to realize that the name element's values are a cache of the last time the document was connected to the database, and the indices specified must be used to connect the field mappings with the columns in the data source. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.14.7 column (Index of Column Containing Unique Values for Record)

This element specifies the column within the specified external data source that contains unique data for the current record within that data source. This element shall be used in conjunction with the uniqueTag element (§17.14.35) to maintain a relationship between a specific record within an external data source and a given source or merged document. The val attribute on this element shall be interpreted as a zero-based index into the columns specified by the data source, specifying the resulting column as the column in which the uniqueTag element shall be looked up.

If this element specifies a column number which exceeds the number of columns in the specified external data source, then its value shall be ignored.

[*Note*: This information is necessary as part of a mail merge as records can be added or deleted from external data sources, and a means must be provided to maintain record-specific inclusion or exclusion data using the active element (§17.14.1) and the affected external data record when the WordprocessingML document is reconnected to the external data source irrespective of the ordering of the records within the external data source. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the information about a single record in a source document for a mail merge:

```
<w:recipientData>
  <w:active w:val="0" />
  <w:column w:val="12" />
  <w:uniqueTag>258865469</w:uniqueTag>
</w:recipientData>
```

The external data record associated with this information is specified via the column in the external data source corresponding to the column element with a val attribute equal to 12, which contains a row whose value in this column has a value corresponding to the uniqueTag element (§17.14.35) with a val attribute equal to 258865469. This record is not imported into the merged WordprocessingML document as the active (§17.14.1) element associated with the given external data record has a val attribute equal to 0.

In other words, when the specified external data source is connected to, the record within the thirteenth column of the external data source that has the contents 258865469, and not populate mapped fields in a merged document with data from that record. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	[*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.14.8 connectString (Data Source Connection String)

This element specifies the connection string used to reconnect to an external data source. The string within this element's val attribute shall contain the connection string that the hosting application shall pass to a external data source access application to enable the WordprocessingML document to be reconnected to the specified external data source.

[*Note*: This string is generally comprised of a series of name/value pairs, delimited by semicolons, determined by the data source access application and the external data source that is accessed. *end note*]

If this string is omitted, then no legacy connection string shall be associated with this mail merge.

This connection string should be ignored under the following conditions:

- The udl element (§17.14.34) is present within the mail merge data
- The dataType element (§17.14.10) is set to `native`
- The current application is able to use the information contained in the odso element (§17.14.25) to access the data source

[*Guidance*: In this case, using the connection string in the udl element provides an equal or greater amount of connection information for the mail merge data source for clients which support it. *end guidance*]

[*Example*: Consider a merged WordprocessingML document that has been connected to an external data source for the purposes of a mail merge. The following WordprocessingML fragment represents the legacy connection string used to connect to the external data source when the merged WordprocessingML document is reopened:

```
<w:connectString w:val="Provider=Example;Password=Test;User ID=readonly;…" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The connectString element specifies that the string `Provider=Example;Password=Test;User ID=readonly;…` must be used to enable the given WordprocessingML document to be reconnected to the specified external data source. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.9 dataSource (Data Source File Path)

This element specifies the relationship whose target is the location of the external data source to be connected to a given WordprocessingML document to perform the mail merge (for a source document) or to find the associated field data (for a merged document).

If this element is omitted, then no file location is specified for the data source for the current mail merge. If no relationship exists with the given relationship ID, or this relationship is not of type http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeSource then this document shall be considered non-conformant.

The data source location can also be ignored under the following conditions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- The src element (§17.14.30) is present within the mail merge data
- The dataType element (§17.14.10) is set to `native`
- The current application is able to use the information contained in the odso element (§17.14.25) to access the data source

[*Guidance*: In this case, using the data source file path in the src element provides an equal or greater amount of information for the mail merge data source for clients which can consume it. *end guidance*]

[*Example*: Consider a WordprocessingML source document containing the following mail merge data:

```
<w:mailMerge>
  …
  <w:dataSource r:id="rId1" />
  …
</w:mailMerge>
```

This mail merge's dataSource element specifies via its r:id attribute value of `rId1` that the external data source to be connected to the given WordprocessingML document is the data source targeted by the relationship whose Id attribute is equal to `rId1`. If we examine the corresponding relationship part item for the setting part, as follows:

```
<Relationships>
  <Relationship Id="rId1"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeSource"
Target="file:///c:/example_file.mdb" TargetMode="External" />
</Relationships>
```

Since the relationship whose Id attribute value is `rId1` specifies the source file path for the data source, that data source effectively specifies a file path of `c:\example_file.mdb`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.14.10 dataType (Data Source Type)

This element specifies the type of external data source to be connected to via the Dynamic Data Exchange (DDE) system (such as a spreadsheet or database), or the alternative method of data access if the Dynamic Data Exchange system is not used. This setting is purely a suggestion of the data source access mechanism which shall be used, and can be ignored in favor of an alternative mechanism if one is present.

[*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document:

```
<w:dataType w:val="odbc" />
```

The dataType element's val attribute is equal to odbc, specifying that the given merged WordprocessingML document has been connected to an external data source via ODBC. *end example*]

Attributes	Description
val (Value)	Specifies the exact type of external data source to which a given merged WordprocessingML document is to be connected. [*Example*: Consider the following WordprocessingML fragment for a mail merge document. `<w:dataType w:val="database" />` The val attribute is equal to `database`, specifying that the given WordprocessingML document has been connected to a database via the Dynamic Data Exchange (DDE) system. *end example*] The possible values for this attribute are defined by the ST_MailMergeDataType simple type (§17.18.52).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMergeDataType) is located in §A.1. *end note*]

17.14.11 destination (Merged Document Destination)

This element specifies what the result which shall be generated when a mail merge is carried out on a given WordprocessingML source document. In other words, this element is used to specify what is to be done with the merged documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.

If this element is omitted, then the default destination of merged documents shall be assumed to be of type `newDocument`.

[*Note*: The aspects of the mail merge outside of connecting to an external data source and populating the fields within a given merged document with external data from the specified external data source are not specified by ISO/IEC 29500.

For example, if a given merged WordprocessingML document contains a destination element with its val attribute equal to `email`, the hosting application can surface a user interface specific to creating emails with the data resulting from populating fields within a given merged WordprocessingML document with external data from the specified external data source. WordprocessingML only provides a flag (via the destination element) to tell the hosting application to surface this user interface. *end note*]

[*Example*: Consider a WordprocessingML source document containing the following WordprocessingML:

```
<w:mailMerge>
  <w:destination w:val="newDocument" />
  …
</w:mailMerge>
```

The destination element's val attribute is set to `newDocument`, specifying that when the mail merge is carried out, the source document must be used to generate a specified number of new documents, which can be handled as appropriate. *end example*]

Attributes	Description
val (Mail Merge Merged Document Type)	Specifies the type of merged documents which shall be the result of carrying out a mail merge on a given source WordprocessingML document. [*Example*: Consider the WordprocessingML mail merge data specified as follows: `<w:destination w:val="email" />` This specifies that a given merged WordprocessingML document is used by the hosting application to generate e-mails containing the static contents of the merged document as well as external data populated into mapped fields. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_MailMergeDest simple type (§17.18.53).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMergeDest) is located in §A.1. *end note*]

17.14.12 doNotSuppressBlankLines (Remove Blank Lines from Merged Documents)

This element specifies how an application performing the mail merge shall handle blank lines in the merged documents resulting from the mail merge. Typically, when a mail merge is performed, any blank lines which result from lines whose sole contents are merge fields with no content are removed from the merged document in order to prevent extraneous blank lines from appearing in the merged documents. When this element is present, the merged documents which are generated from the mail merge shall not have any blank lines removed before they are sent to their destination format.

If this element is omitted, the merged documents generated from this mail merge shall have all blank lines suppressed if they consist of only merge fields with values consisting of empty strings.

[*Example*: Consider a WordprocessingML document containing a single WordprocessingML field calling for `Test` data as seen in the first column of the table below. If the current record in the mail merge data source contains an empty string for the `Test` column, the resulting merged document would be displayed as follows depending on the setting for the doNotSuppressBlankLines element:

Source Document	`<w:doNotSuppressBlankLines val="true" />`	`<w:doNotSuppressBlankLines val="false" />`
One Two {MERGEFIELD "Test" } Three	One Two Three	One Two Three

With this element set to a value of `true`, the blank lines in the resulting document must not be suppressed when the resulting merged documents are created. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.14.13 dynamicAddress (Use Country-Based Address Field Ordering)

This element specifies that the contents of the `AddressBlock MERGEFIELD` field shall be dynamically ordered based on the country associated with the current record or if the country-invariant version of the address field shall be used in its place. [*Rationale*: When a source document is combined with the contents of a data source in order to produce multiple merged documents, it is often necessary to use an address form specific to the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

destination country for each particular record in the data source, rather than one static address form for all records. *end rationale*] If this element is set to `true`, then the mail merge shall use an address form suited to the country associated with the current record in the external data source.

If this element is omitted, then the form of the address shall be dynamically determined based on the country specified in the current record.

[*Example*: Consider a merged WordprocessingML document that is specified to *not* dynamically create the address field order based on the country associated with the current record. This requirement might be specified using the following WordprocessingML:

```
<w:fieldMapData>
  …
  <w:dynamicAddress w:val="off" />
</w:fieldMapData>
```

The dynamicAddress element is set to a value of `off`, specifying that the dynamic address format must not be used when performing a mail merge with the specified data source. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.14.14 fHdr (First Row of Data Source Contains Column Names)

This element specifies that a hosting application shall treat the first row of data in the specified external data source as a header row containing the names of each column in the data source, rather than data to populate mapped fields in a merged document. When present, this information shall not change the indices specified in the recipientData elements (§17.14.27), but shall indicate that the first row is not part of the mail merge when it is performed.

If this element is omitted, then the first row of the data source shall not be considered a header row when a mail merge is performed.

[*Example*: Consider a WordprocessingML source document that has been connected to an external data source whose first row of data is not data the hosting application is to populate mapped fields with, but rather contains column names for each column in the data source. This setting on the data source is specified using the following fragment of WordprocessingML:

```
<w:fHdr w:val="on" />
```

The fHdr element specifies that the data source's first row is a header row, rather than regular data to be used in the mail merge. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.14.15 fieldMapData (External Data Source to Merge Field Mapping)

This element specifies how a column specified in the external data source that has been connected to a WordprocessingML document shall be mapped to the pre-defined MERGEFIELD fields (§17.16.5.35) within the given merged document's contents. Each instance of a fieldMapData element contains the information needed to map one column in the external data source to a single type of pre-defined MERGEFIELD field for the purposes of the mail merge in the current document.

[*Example*: Consider a single merged document. The WordprocessingML below demonstrates the mapping of the Country column from the external data source to the predefined WordprocessingML Country or Region merge field when the merged document is populated with external data as part of a mail merge:

```
<w:odso>
  …
  <w:fieldMapData>
    <w:type w:val="dbColumn" />
    <w:name w:val="Country" />
    <w:mappedName w:val="Country or Region" />
    <w:column w:val="9" />
    …
  </w:fieldMapData>
</w:odso>
```

The fieldMapData element specifies the mapping between the external data source and a single merge field as follows: the child elements specify that the tenth column in the data source, last titled Country in the specified external data source when the connection was last made is to be mapped to the predefined WordprocessingML merge field calling for Country or Region data.

With the fieldMapData element configured as such, an application can be used in conjunction with this WordprocessingML document to populate the document with data mapped from the specified external data source to fields within the merged document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OdsoFieldMapData) is located in §A.1. *end note*]

17.14.16 headerSource (Header Definition File Path)

This element specifies the location of a file that contains the column header information which shall be used when connecting to an external data source that does not have column header data specified. Specifically, this element specifies a file that corresponds with the aforementioned external data source. [*Note*: Column headers are needed to enable a hosting application to associate external data source's columns to fields via the fieldMapData element (§17.14.15).

If this element is omitted, then the column header definition data is not specified in an external file and shall be retrieved from the primary data source associated with the mail merge.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a WordprocessingML merged document containing the following WordprocessingML:

```
<w:settings>
  …
  <w:headerSource r:id="rId2" />
  …
</w:settings>
```

This mail merge's headerSource element specifies via its r:id attribute value of `rId2` that the external data source to be used for the column header information for the given WordprocessingML document is the data source targeted by the relationship whose Id attribute is equal to `rId2`. If we examine the corresponding relationship part item for the setting part, as follows:

```
<Relationships>
  <Relationship Id="rId2"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeSource"
Target="file:///c:/headerData.txt" TargetMode="External" />
</Relationships>
```

Since the relationship whose Id attribute value is `rId2` specifies the column header data file path for the data source effectively specifies a file path of `c:\headerData.txt`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.14.17 lid (Merge Field Name Language ID)

This element specifies the language ID for the language which was used to generate the merge field name which was associated with a given column in the data source, as specified by the fieldMapData element (§17.14.15). This element specifies that when this field mapping is processed by an application, it shall interpret the merge field name as the name for the specified language.

If this element is omitted, then the mapped field names specified in the current document can be interpreted using any method desired by the consuming application (i.e. no language data is included with the field mapping information).

[*Example*: Consider the following WordprocessingML fragment for a field mapping for a document to be merged to an external data source. If the merge field name stored in the file corresponds with the U.S. English version of the merge field names, that information would be stored as follows:

```
<w:fieldMapData>
  <w:name w:val="Title" />
  <w:mappedName w:val="Courtesy Title" />
  <w:column w:val="3" />
  <w:lid w:val="en-US" />
  …
</w:fieldMapData>
```

The lid element specifies that the mapping of the data contained in the fourth column of external data source named `Title` to the WordprocessingML of 'Courtesy Title', must be associated with the U.S. English language as specified by the val attribute equal to `en-US`. *end example*]

Attributes	Description
val (Language Code)	Specifies an identifier for a specific language. This code is interpreted in the context of the parent XML element. [*Example*: Consider an object which must specify the English(Canada) language. That object would use an identifier of `en-CA` to specify this language. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lang) is located in §A.1. *end note*]

17.14.18 linkToQuery (Query Contains Link to External Query File)

This element specifies that the current WordprocessingML document's query string, stored in the query element (§17.14.26) and used to specify the data to be imported from the external data source, actually contains a reference to an external query file which contains the actual query data to be used against the specified external data source for the mail merge. This query shall mimic a SQL query and be of the following form: `SELECT * FROM <query file path>`.

If this element is omitted, then the query specified for the data source attached to the current document shall be assumed to not be a query containing a link to an external file.

[*Example*: Consider a mail merge source document that uses the linkToQuery element to specify that the query used is stored in the specified external data source as follows:

```
<w:mailMerge>
  …
  <w:linkToQuery />
  <w:query w:val="SELECT * FROM C:\queryExample.txt" />
  …
</w:mailMerge>
```

The linkToQuery element specifies that the query string stored in the query element (§17.14.26) is actually just a reference to an externally stored query file, in this case, an external query file stored at `c:\queryExample.txt`. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.14.19 mailAsAttachment (Merged Document To E-Mail Attachment)

This element specifies that, after importing external data into fields to generate a series of destination WordprocessingML documents as e-mails, the resulting documents should be emailed as an attachment rather than the body of the actual e-mail.

If this element is omitted, then its value shall be assumed to be `false` (i.e. the destination source is not an e-mail attachment). If the destination element (§17.14.11) specifies that the merged document destination is not `email`, then this element shall be ignored.

[*Example*: Consider a merged WordprocessingML document that has been connected to an external data source containing three records and that contains the following WordprocessingML in its mail merge properties as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:mailMerge>
  …
  <w:destination w:val="email" />
  <w:mailAsAttachment />
  …
</w:mailMerge>
```

After the external data has been imported into the merged document's respective merge fields, three emails is generated (as specified by the destination element (§17.14.11) with a val attribute of `email`, each an attachment consisting of one of the three documents resulting from the mail merge result (rather than just including the merged document as the body of the email). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.14.20 mailMerge (Mail Merge Settings)

This element specifies all of the mail merge information for a document that has been connected to an external data source as part of a mail merge operation.

The document which contains this mail merge data can be of one of two types:

- A *source document*, the document which contains all of the information for the mail merge, and is used in conjunction with an application to connect to an external data source and create one document for each record in that data source.
- A *merged document*, a document which contains all of the information for the mail merge as well as a reference to a single specific record which shall be used to populate the values of all of the merge fields in that document.

The information in this element shall contain all data needed to connect to a data source and populate any merge fields in the document with data from that data source.

[*Example*: Consider the following WordprocessingML fragment for a document which is part of a mail merge:

```
<w:mailMerge>
  …
  <w:dataType w:val="spreadsheet" />
  <w:query w:val="SELECT * FROM `Sheet1$`" />
  <w:dataSource r:id="rId1" />
  …
</w:mailMerge>
```

Here, the dataType (§17.14.10) and dataSource (§17.14.9) elements specify that the given document must be connected to the external data source referenced by the relationship whose id value is equal to `rId1`. While connected to the external data source, the document together with a hosting application can extract data from the external data source as specified by the query (§17.14.26) element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMerge) is located in §A.1. *end note*]

17.14.21 mailSubject (Merged E-mail or Fax Subject Line)

This element specifies the text which shall appear in the subject line of the e-mails or faxes that result after the actions of a mail merge have imported external data into fields within a merged WordprocessingML document whose destination, as specified in the destination element (§17.14.21), is `email` or `fax`.

If this element is omitted, then no subject line text shall be associated with each merged document produced via a mail merge using the specified mail merge data. If the destination element (§17.14.11) specifies that the merged document destination is not `email` or `fax`, this element shall be ignored.

[*Example*: Consider a merged WordprocessingML document containing fields and the following WordprocessingML as part of its mail merge data:

```
<w:mailMerge>
  …
  <w:destination w:val="email" />
  <w:mailSubject w:val="Example Subject Line" />
  …
</w:mailMerge>
```

The mailSubject element specifies that after the specified external data has been imported into the specified fields in the merged document, each record merged must result in a single e-mail message, each with their subject line reading `Example Subject Line`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.22 mainDocumentType (Source Document Type)

This element specifies the document type of a given WordprocessingML source document.

If this element is omitted, then its value shall be assumed to be `formLetters`.

[*Note*: This element is generally used in conjunction with the behavior of an application to customize aspects of the mail merge user interface and experience independent of the WordprocessingML file format. For example, if a given WordprocessingML merged document contains a mainDocumentType element with its val attribute equal to `envelopes`, the hosting application can surface a piece of user interface specific to creating envelopes when the given document is opened.

In addition, what a hosting application does with the documents that result from importing external data into specified fields can be determined based on the mainDocumentType element, but other than this, is independent of a given merged document's WordprocessingML. For example, if a given merged WordprocessingML document contains a mainDocumentType element with its val attribute equal to `email`, the hosting application can call a email service after importing external data into specified fields, in order to generate emails containing the resulting documents.

WordprocessingML simply provides the mainDocumentType that can serve as a trigger for an application to surface user interface specific to a document type of mail merge. *end note*]

[*Example*: Consider the WordprocessingML below:

```
<w:mailMerge>
  <w:mainDocumentType w:val="formLetters" />
  …
</w:mailMerge>
```

In this example, the source document is of the `formLetters` type, as specified by the mainDocumentType element's val attribute being equal to `formLetters`. *end example*]

Attributes	Description
val (Mail Merge	Specifies the type of source document which is specified by the given WordprocessingML

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Source Document Type)	document. [*Example*: Consider the WordprocessingML below: `<w:mainDocumentType w:val="formLetters" />` This WordprocessingML specifies that a given source document is a `formLetters` document. This setting implies nothing about the file, but can be interpreted by an application as desired. *end example*] The possible values for this attribute are defined by the ST_MailMergeDocType simple type (§17.18.54).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMergeDocType) is located in §A.1. *end note*]

17.14.23 mappedName (Predefined Merge Field Name)

This element specifies the predefined WordprocessingML `MERGEFIELD` field name which shall be mapped to the column number specified by the column element (§17.14.6) within this field mapping. [*Guidance*: This element allows the current column from the specified data source to be mapped to a predefined field name, allowing applications to have one standard set of field names to use regardless of the data source column names, for example, to create the address formats to place into an `ADDRESSBLOCK` field. *end guidance*]

If this element is omitted, then the current data source column mapping shall not have a predefined merge field name mapped to its contents, and shall only be referenced via the data source column name specified by the name element (§17.14.24) when referenced by one or more `MERGEFIELD` fields. If the application does not have a predefined merge field whose name matches the name specified using the val attribute, then this element can be ignored.

[*Example*: Consider the following WordprocessingML fragment, representing two columns from an external data source which have been mapped to the built-in fields `First Name` and `Last Name`, respectively:

```
<w:fieldMapData>
  <w:name w:val="Column Name A" />
  <w:mappedName w:val="First Name" />
  <w:column w:val="0" />
  …
</w:fieldMapData>
<w:fieldMapData>
  <w:name w:val="Column Name B" />
  <w:mappedName w:val="Last Name" />
  <w:column w:val="1" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    …
</w:fieldMapData>
```

The first and second columns, specified by the column element values of `0` and `1` respectively, specify that the predefined WordprocessingML field names `First Name` and `Last Name` are mapped to the columns of the external data source, and the data source names for those columns are `Column Name A` and `Column Name B`, respectively.

Therefore, if `MERGEFIELD` fields calling for `First Name` and `Last Name` are inserted in a WordprocessingML document connected to the external data source with the field mappings specified above, when the mail merge takes place, the data from the first and second column populates the fields calling for `First Name` and `Last Name` data within the merged WordprocessingML document. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.14.24 name (Data Source Name for Column)

This element specifies the column name within a given external data source for the column whose index is specified via the column element (§17.14.6). This data source name provides a column name which shall be used to map a specific MERGEFIELD field in the document, as specified by the parent field mapping data. The val attribute specifies the name of this column in the data source when the connection is initially established, which is then used permanently to link columns in the database to MERGEFIELD fields in the document.

If this element is omitted, no data source name is provided for the current column.

[*Example*: Consider a source document that is connected to an external data source with three columns. Within this external data source, these are three columns are ordered and titled as follows: first, middle, and last, respectively. The following WordprocessingML specifies that when this document was connected to the data source, these columns were ordered in this manner:

```
<w:fieldMapData>
  …
  <w:name w:val="first" />
  <w:column w:val="0" />
</w:fieldMapData>
<w:fieldMapData>
  …
  <w:name w:val="middle" />
  <w:column w:val="1" />
</w:fieldMapData>
<w:fieldMapData>
  …
  <w:name w:val="last" />
  <w:column w:val="2" />
</w:fieldMapData>
```

The WordprocessingML above demonstrates that the column name first must be associated with the first column in the external database by specifying a column element with its val attribute equal to 0. In addition, the column name middle must be associated with the second column in the external database by specifying a column element with its val attribute equal to 1. Finally, the column name last must be associated with the third column in the external database by specifying a column element with its val attribute equal to 2. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.25 odso (Office Data Source Object Settings)

This element specifies a group of additional settings for the mail merge information which comprise an extension to the standard settings stored with a mail merge which performs two functions:

- First, it provides additional information about the mail merge data source, specifically: information about how to map the columns in the data source to MERGEFIELD fields and information about records which shall be included and excluded when creating merged documents, and column delimiters used in text data sources. This information can be used regardless of the value of the dataType element (§17.14.10) when it is present.
- Second, it provides an alternate set of connection information which should be used when the dataType element (§17.14.10) specifies a value of native. This alternate connection string provides additional connection information for applications which choose to support the ODSO connection string syntax. If the dataType element (§17.14.10) specifies that the data type of the current mail merge is not native, then the second group of settings specified within this element shall be ignored in favor of their non-ODSO equivalents.

[*Example*: Consider the WordprocessingML for a source document whose mail merge includes mail merge information including ODSO settings as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:odso>
  <w:udl w:val="…" />
  <w:table w:val="Sheet1$" />
  <w:src r:id="rId1" />
  <w:colDelim w:val="9" />
  <w:fHdr w:val="1" />
  <w:fieldMapData>
    <w:type w:val="dbColumn" />
    <w:name w:val="Title" />
    <w:mappedName w:val="Courtesy Title" />
    <w:lid w:val="en-US" />
  </w:fieldMapData>
  <w:recipientData r:id="rId2" />
  …
</w:odso>
```

The odso element and its child elements provide all of the information specified above that is needed to carry out a mail merge with the current document. This includes alternate connection information using the udl, table, and src elements, and additional mail merge information in the other child elements. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Odso) is located in §A.1. *end note*]

17.14.26 query (Query For Data Source Records To Merge)

This element contains the Structured Query Language string (as defined by the normative reference in§3) that shall be run against the specified external data source to return the set of records from the external data which shall be imported into merged WordprocessingML documents when the mail merge operation is performed.

If this element is omitted, then no query shall be associated with the current data source.

[*Example*: Consider a WordprocessingML document that has been connected to an external database. In addition, consider that the data specifies that the table within the database titled Documentation shall be the specific table whose data is imported. This shall be specified in WordprocessingML as follows:

```
<w:query w:val="SELECT * FROM Documentation" />
```

The query element specifies the syntax for the data source query via its val attribute. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.27 recipientData (Data About Single Data Source Record)

This element specifies information about a single record within an external data source. If a record shall be merged into a merged document, then no information is needed about that record within this part. However, if a given record shall not be merged into a merged document, then the value of the unique key for that record shall be stored within the uniqueTag element as a child of this element (along with the active element) to indicate this exclusion.

[*Note*: This mapping is necessary in place of simply using the element order to correspond to the record indices in the external data source, as records can be added or deleted from external data sources, and a means must be provided to maintain WordprocessingML external record specific data like that specified in the active element (§17.14.1) and the corresponding external data record when the WordprocessingML document is reconnected to the external data source; irrespective of the ordering of the records within the external data source. In other words, this element, and its child elements enable merged WordprocessingML documents to maintain the relationship between the records within an external data and record specific WordprocessingML parameters. *end note*]

[*Example*: Consider a merged WordprocessingML document that:

1. Has been connected to a specified external data source containing three records; and
2. Has been configured by the hosting application to not populate a merged document with the record pertaining to John Smith in the external data source.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Consider also that the first time the given WordprocessingML document was connected to the external data source, John Smith's record was in the second record in the data source.

When this merged document is connected to the external data source the recipientData element can be used to store the number and value of the column containing the unique key for each data record within the external data source including John Smith's. This setting is represented using the following WordprocessingML to use the hash codes within the recipientData element to uniquely identify the three records within the external data source.

```
<w:recipientData>
  <w:column w:val="1" />
  <w:uniqueTag>1408613399</w:uniqueTag>
</w:recipientData>
<w:recipientData>
  <w:active w:val="0" />
  <w:column w:val="1" />
  <w:uniqueTag>870254691</w:uniqueTag>
</w:recipientData>
<w:recipientData>
  <w:column w:val="1" />
  <w:uniqueTag>1107777181</w:uniqueTag>
</w:recipientData>
```

Here, the first, second (John Smith record), and third records within the specified data source whose unique key values are 1408613399, 870254691, and 1107777181 have been associated with with recipient data via the active element to specify that the record associated with the given record (John Smith's record) must not be used to populate a merged WordprocessingML document.

With these association in place, if a fourth record is added to the given external data source above John Smith's record, when the given merged WordprocessingML document is reconnected to the external data source, the hosting application still knows that John Smith's record must not be used to populate a merged WordprocessingML document as it is associated via its unique key value and is not dependent on the given record's ordinal position within the external data source. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RecipientData) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.14.28 recipientData (Reference to Inclusion/Exclusion Data for Data Source)

This element shall specify a reference to the part which contains data about whether the set of records in the associated data source have been explicitly included or excluded from the specified mail merge. Only those records which must not be used to generate merged WordprocessingML documents shall be stored within the referenced part, as all records shall be merged by default as part of the mail merge operation. [*Guidance*: Applications can choose to store only those records which are excluded for efficiency, or a list of all records in order to determine which set of records were added/removed between mail merge operations. *end guidance*]

 ©ISO/IEC 2016 – All rights reserved

[*Rationale*: When defining a mail merge, it is possible that a user wishes to connect to a specified data source, but specify only a subset of the records returned by the query specified by the query element (§17.14.26) which must be merged as part of the mail merge operation. This element allows applications to utilize a separate part to store this information, either the shared part defined by ISO/IEC 29500, or an application-defined part as needed. *end rationale*]

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeRecipientData, is not present, or does not have a TargetMode attribute value of `Internal`, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it can be ignored.

ISO/IEC 29500 defines one shared mechanism for storing this data: using the Mail Merge Recipient Data part. This mechanism shall be used if the associated data source has a column which can be used as the unique key. However, when using data sources which do not have a unique key, applications can store their own part (of an application-defined content type) using this relationship.

[*Example*: Consider a WordprocessingML document which is a mail merge source document, containing inclusion/exclusion data for the data source. The document settings part would contain the mail merge data:

```
<w:settings>
  …
  <w:mailMerge>
     …
     <w:odso>
       …
       <w:recipientData r:id="recipient1" />
     </w:odso>
  </w:mailMerge>
</w:settings>
```

The recipientData element specifies that the external content targeted by the relationship with an ID of `recipient1` contains the recipient inclusion/exclusion data for the mail merge operation. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="recipient1" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeRecipientData,"
Target="recipientData.xml" />
  …
</Relationships>
```

The corresponding relationship part item shows that the file containing this data is located next to the main document and is named `recipientData.xml`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.14.29 recipients (Inclusion/Exclusion Data for Data Source)

This element specifies all of the inclusion/exclusion data for the contents of the specified mail merge data source. It is the root element for the Mail Merge Recipient Data part.

[*Example*: Consider a document which is the source document for a mail merge operation. If two records of the three specified by the data source were excluded from the mail merge, the resulting recipient data part would appear as follows:

```
<w:recipients>
  <w:recipientData>
    <w:active w:val="false"/>
    …
  </w:recipientData>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:recipientData>
      <w:active w:val="false"/>
      …
    </w:recipientData>
    <w:recipientData>
      …
    </w:recipientData>
  </w:recipients>
```

The recipients element contains all of the recipient inclusion/exclusion data for this mail merge document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Recipients) is located in §A.1. *end note*]

17.14.30 src (ODSO Data Source File Path)

This element specifies the relationship whose target is the location of the external data source to be connected to a given WordprocessingML document to perform the mail merge (for a source document) or to find the associated field data (for a merged document) when the merge type, specified using the dataType element (§17.14.10), is set to `native`.

If this element is omitted, then no file location is specified for the data source for the current mail merge. If no relationship exists with the given relationship ID, or this relationship is not of the Mail Merge Data Source relationship type, then this document shall be considered non-conformant.

The data source location is only used under the following conditions:

- The dataType element (§17.14.10) is set to `native`
- The current application is able to use the information contained in the odso element (§17.14.25) to access the data source

[*Guidance*: In this case, using the data source file path in the src element provides an equal or greater amount of information for the mail merge data source for clients which can consume it. *end guidance*]

[*Example*: Consider a WordprocessingML source document containing the following mail merge data:

```
  <w:odso>
    …
    <w:src r:id="rId1" />
    …
  </w:odso>
```

This mail merge's src element specifies via its r:id attribute value of `rId1` that the external data source to be connected to the given WordprocessingML document is the data source targeted by the relationship whose Id

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

attribute is equal to `rId1`. If we examine the corresponding relationship part item for the setting part, as follows:

```
<Relationships>
  <Relationship Id="rId1"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/mailMergeSource"
Target="file:///c:/example_file.mdb" TargetMode="External" />
</Relationships>
```

Since the relationship whose Id attribute value is `rId1` specifies the source file path for the data source, that data source effectively specifies a file path of `c:\example_file.mdb`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.14.31 table (Data Source Table Name)

This element specifies the particular set of data that a source or merged WordprocessingML document shall be connected to within an external data source containing multiple data sets. In other words, when connecting to a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

WordprocessingML document to an external data source that can have more than one repository of data within it, such as a database with multiple tables or a spreadsheet with multiple worksheets, this element is used to distinguish the specific table or spreadsheet from which data is imported from within the external data source.

[*Example*: Consider a WordprocessingML document that has been connected to database containing two tables named `Table One` and `Table Two`, respectively. To specify that the mail merge must import data from `Table One` into the WordprocessingML document, this requirement would be specified using the following WordprocessingML:

```
<w:odso>
  …
  <w:table w:val="Table One" />
  …
</w:odso>
```

The table element with a value of `Table One` specifies that the external data must be retrieved from this table in the data source. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.14.32 type (ODSO Data Source Type)

This element specifies the type of external data source to be connected to via as part of the ODSO connection information for this mail merge. This setting is purely a suggestion of the data source type which is being used for this mail merge, and can be ignored in favor of an alternative mechanism if one is present.

[*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document:

```
<w:type w:val="database" />
```

The type element's val attribute is equal to `database`, specifying that the given merged WordprocessingML document has been connected to an external data source via the ODSO settings, and that the resulting data source was a database. *end example*]

Attributes	Description
val (Data Source Type Value)	Specifies the type of an external data source used for a mail merge operation. [*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document: `<w:type w:val="text" />` The val attribute is equal to `text`, specifying that the given merged WordprocessingML document has been connected to an external data source via the ODSO settings, and that the resulting data source was a text file. *end example*] The possible values for this attribute are defined by the ST_MailMergeSourceType simple type (§17.18.56).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMergeSourceType) is located in §A.1. *end note*]

17.14.33 type (Merge Field Mapping)

This element specifies if a given mail merge field has been mapped to a column in the given external data source or not.

If this element is omitted, then the field mapping shall be considered to be of type `null` (i.e. not mapped).

[*Example*: Consider the WordprocessingML for a single field mapping within a mail merge source document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:odso>
  …
  <w:fieldMapData>
    <w:type w:val="dbColumn" />
    <w:name w:val="Country" />
    <w:mappedName w:val="Country or Region" />
    <w:column w:val="9" />
    …
  </w:fieldMapData>
</w:odso>
```

In this example, the `country` column within the given external data source must be mapped to the mail merge field `Country or Region`, as specified by the type element's val attribute being equal to `dbColumn`. *end example*]

Attributes	Description
val (Merge Field Mapping Type)	Specifies if the given mail merge field has been mapped to a column in the given external data source (i.e. if the merge field mapping is active or not). [*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document: `<w:type w:val="null" />` In this example, the given mail merge field must not be mapped to a column in the given external data source, as specified by the type element's val attribute being equal to `null`. *end example*] The possible values for this attribute are defined by the ST_MailMergeOdsoFMDFieldType simple type (§17.18.55).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MailMergeOdsoFMDFieldType) is located in §A.1. *end note*]

17.14.34 udl (UDL Connection String)

This element specifies the Universal Data Link (UDL) connection string used to reconnect to an external data source. The string within this element's val attribute shall contain the connection string that the hosting application shall pass to a external data source access application to enable the WordprocessingML document to be reconnected to the specified external data source.

If this string is omitted, then no UDL connection string shall be associated with the ODSO data for this mail merge.

This connection string is only used under the following conditions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- The dataType element (§17.14.10) is set to `native`
- The current application is able to use the information contained in the odso element (§17.14.25) to access the data source

[*Guidance*: In this case, using the connection string in the udl element provides an equal or greater amount of information for the mail merge data source for clients which can consume it. *end guidance*]

[*Example*: Consider a merged WordprocessingML document that has been connected to an external data source for the purposes of a mail merge. The following WordprocessingML fragment represents the legacy connection string used to connect to the external data source when the merged WordprocessingML document is reopened:

```
<w:udl w:val="Provider=Example;Password=Test;User ID=readonly;…" />
```

The udl element specifies that the string `Provider=Example;Password=Test;User ID=readonly;…` must be used to enable the given WordprocessingML document to be reconnected to the specified external data source. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.14.35 uniqueTag (Unique Value for Record)

This element specifies the contents of a given record within the specified external data source, in the column containing unique data for every record within the external data source. given external data source. This element is used in conjunction with the column (§17.14.6) element to maintain a relationship between the records within an external data source and a given merged WordprocessingML document.

The contents of this attribute shall be the base64-encoded value of the unique tag value as specified by the data source.

[*Note*: This information is necessary as part of a mail merge as records can be added or deleted from external data sources, and a means must be provided to maintain record-specific inclusion or exclusion data using the active element (§17.14.1) and the affected external data record when the WordprocessingML document is reconnected to the external data source irrespective of the ordering of the records within the external data source. *end note*]

[*Example*: Consider the following WordprocessingML fragment for the information about a single record in a source document for a mail merge:

```
<w:recipientData>
  <w:active w:val="0" />
  <w:column w:val="12" />
  <w:uniqueTag>258865469</w:uniqueTag>
</w:recipientData>
```

The external data record associated with this information is specified via the column in the external data source corresponding to the column element with a val attribute equal to 12, which contains a row whose value in this column has a value corresponding to the uniqueTag element with a val attribute equal to 258865469. This record is not imported into the merged WordprocessingML document as the active (§17.14.1) element associated with the given external data record has a val attribute equal to 0.

In other words, when the specified external data source is connected to, the record within the thirteenth column of the external data source that has the contents 258865469, and not populate mapped fields in a merged document with data from that record. *end example*]

The possible values for this element are defined by the W3C XML Schema CT_Base64Binarydatatype.

17.14.36 viewMergedData (View Merged Data Within Document)

This element specifies that a specific merged document shall display the data from the specified external data source where merge fields have been inserted. The activeRecord element (§17.14.2) is used to specify which record within the external data source is to have its applicable data displayed where applicable within the WordprocessingML merged document.

If the activeRecord element is not present in the WordprocessingML for the document with the viewMergedData's val attribute equal to true, the hosting application can behave as if the acitveRecord

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

element's val attribute was equal to 1. This element is ignored if the viewMergedData (§17.14.36) element is not present or present with a val attribute equal to Off, 0, or false.

[*Example*: Consider a merged WordprocessingML document containing two WordprocessingML fields calling for Courtesy Title and Last Name data and a sample text paragraph. Also, note that the external data source this merged document is connected to contains two records, both containing name and address information, with the first record pertaining to Mr. Doe, and the second pertaining to Ms. Smith.

This table below illustrates the necessary WordprocessingML to display applicable data from the specified external data source within the merged document where fields have been inserted:

`<w:viewMergedData val="0"/>`	`<w:viewMergedData val="1" />` `<w:activeRecord w:val="1" />`	`<w:viewMergedData val="0" />` `<w:activeRecord w:val="1" />`
Dear {MERGEFIELD "Courtesy Title" \m} {MERGEFIELD "Last Name" \m}, Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,	Dear Mr. Doe: Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,	Dear {MERGEFIELD "Courtesy Title" \m} {MERGEFIELD "Last Name" \m}, Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sample letter text. Sincerely,

The viewMergedData element specifies that the specified record in the external data source must be displayed in place of merge fields in the current document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15 Settings

Within a WordprocessingML document, *settings* specify stored preferences which shall be used when processing the contents of the document. These settings are typically divided into three categories:

- *Document Settings* - settings which influence the appearance and behavior of the current document, as well as store document-level state.
- *Compatibility Settings* - settings which tell applications to perform behaviors which are designed to maintain visual output of previous word processing applications. These settings are for backward compatibility and are all ignorable.
- *Web Settings* - settings which affect how this document shall be handled when it is saved as HTML. These settings exist primarily for backward compatibility reasons and are all ignorable.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The first two groups are stored in the Document Settings part, and the last group is stored in the Web Settings part.

17.15.1 Document Settings

The first group of settings stored in WordprocessingML is document settings. These settings specify all document-level properties which affect the handling of the current document.

[*Example*: Consider the following WordprocessingML fragment for the document settings in a WordprocessingML document:

```
<w:settings>
  <w:defaultTabStop w:val="720" />
  <w:characterSpacingControl w:val="doNotCompress" />
</w:settings>
```

The settings element contains all of the document settings for this document. In this case, the two settings applied are automatic tab stop increments of 0.5" using the defaultTabStop element (§17.15.1.25), and no character level whitespace compression using the characterSpacingControl element (§17.15.1.18). *end example*]

17.15.1.1 activeWritingStyle (Grammar Checking Settings)

This element specifies information about the parameters of the grammar checking which was performed on the contents of the current WordprocessingML document. [*Note*: This information can be used as desired by applications; for example, to determine if the current grammar checking state, specified by the proofState element (§17.15.1.65) is sufficient. *end note*]

[*Example*: Consider the following WordprocessingML fragment from the document settings:

```
<w:activeWritingStyle w:lang="en-CA" w:vendorID="64" w:dllVersion="131078"
w:nlCheck="1" w:checkStyle="0" w:appName="testApp" />
```

The activeWritingStyle element's lang attribute specifies that the English (Canada) language setting for grammatical and stylistic checks must be applied; the vendorID attribute specifies information about the vendor associated with the DLL used to perform the grammatical and stylistic checks; the dllVersion attribute specifies the version of this DLL; the nlCheck attribute specifies if natural language checks were performed or not; the checkStyle attribute specifies that the hosting application should allow its grammar engine to check both the grammar and style of the given WordprocessingML document, if that functionality is available; and the appName attribute indicates that an application called `testApp` specified the grammar checking rules of the given WordprocessingML. *end example*]

Attributes	Description
appName (Application Name)	Specifies the name of the application which specified the grammatical settings contained on the attributes for this element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If an application reads these settings and does not understand the value of this attribute, then its settings can be ignored and the application's default settings used instead. [*Example*: Consider the WordprocessingML below: `<w:activeWritingStyle … w:appName="testApp"/>` The appName attribute has a value of `testApp`, specifying that the application called `testApp` specified the grammar checking rules of the given WordprocessingML document. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
checkStyle (Check Stylistic Rules With Grammar)	Specifies if the grammar content checking performed on this document included stylistic rules for the document content. If specified, applications which support this functionality shall check stylistic rules as well as grammatical ones when checking the grammatical content of this document. [*Example*: Consider the WordprocessingML below: `<w:activeWritingStyle … w:checkStyle="false"/>` The checkStyle attribute has a value of `false`, specifying that hosting applications must only check grammatical rules of the given WordprocessingML document. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
dllVersion (Grammatical Check Engine Version)	Specifies the version of the engine that was used to check the grammatical content of the WordprocessingML document. [*Example*: Consider the following WordprocessingML fragment: `<w:activeWritingStyle … w:dllVersion="131078" />` The dllVersion attribute specifies that the writing style DLL version used to check the writing style of is the writing style DLL version associated with the string `131078`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
lang (Writing Style Language)	Specifies the language of the engine used to perform the grammatical content checking. [*Example*: Consider the following WordprocessingML fragment: `<w:activeWritingStyle w:lang="en-CA" …/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The lang attribute has a value of en-CA, therefore the grammatical check language is specified as English (en) and Canada (CA), resulting in use of the English (Canada) grammar checker. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
nlCheck (Natural Language Grammar Check)	Specifies whether the engine that was used to check the grammatical content of the WordprocessingML document performed natural language-based analysis. [*Example*: Consider the WordprocessingML below: `<w:activeWritingStyle … w:nlCheck="1" />` The nlCheck attribute has a value of 1, specifying that the writing style DLL supported natural language analysis . *end example*]. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
vendorID (Grammatical Engine ID)	Specifies a value indicating a unique ID for the writing style engine that was used to check the grammatical content of the WordprocessingML document. [*Example*: Consider the WordprocessingML below: `<w:activeWritingStyle … w:vendorID="64"/>` The vendorId attribute has a value of 64, specifying that the grammatical checker used is identified by the string 64. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WritingStyle) is located in §A.1. *end note*]

17.15.1.2 alignBordersAndEdges (Align Paragraph and Table Borders with Page Border)

This element specifies that paragraph borders specified using the pBdr element (§17.3.1.24) and table borders using the tblBorders element (§17.4.39) shall be adjusted to align with extents of the page border defined using the pgBorders element (§17.6.10) if the spacing between these borders is less than or equal to 10.5 points (one character width) or less from the page border. The presence of this setting shall ensure there are no gaps of one character width or less between adjoining page and paragraph/table borders, as borders which are perfectly aligning shall not be displayed in favor of the intervening page border.

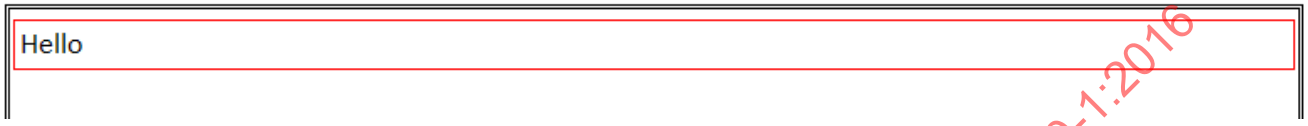
If this element is omitted, then borders shall not be automatically adjusted to prevent gaps of less than one character width. If the page border is not measured from the text extents using a value of `text` in the offsetFrom attribute on the pgBorders element, then it can be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

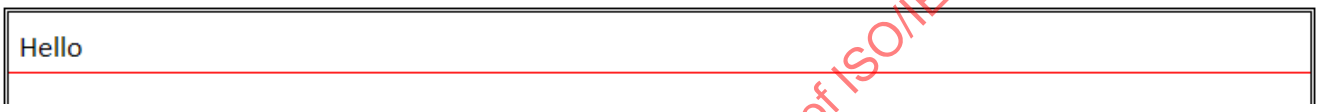
[*Example*: Consider the following WordprocessingML fragment from the document settings:

```
<w:alignBordersAndEdges w:val="true"/>
```

The alignBordersAndEdges element has a value of `true` specifying that borders must be adjusted to prevent gaps of less than one character width. If a document has a page border specified to appear 4 points from the text extents, and within that page a paragraph border specified to appear one point from the text extents, that would normally appear like this:



If this element is present, then those gaps (which are all of three points in width) must be adjusted to ensure that the borders align exactly and the paragraph border is suppressed:



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.3 alwaysMergeEmptyNamespace (Do Not Mark Custom XML Elements With No Namespace As Invalid)

This element specifies whether custom XML markup specified via the customXml element which has no associated namespace shall be treated as an error and moved into a special error namespace (for the purposes of XML schema validation) when the document is opened. If this element is turned on, when an application determines that the current XML markup is in the empty namespace, those elements shall not automatically be moved into an error namespace.

If this element is not present in a WordprocessingML document than custom XML markup which has no associated namespace shall be treated as an error and moved into a special error namespace when the document is opened.

[*Example*: Consider a WordprocessingML document which should not automatically flag empty namespace XML as invalid. This requirement would be specified using the following WordprocessingML:

```
<w:alwaysMergeEmptyNamespace w:val="true"/>
```

The alwaysMergeEmptyNamespace element's val attribute has a value of `true` specifying that custom XML markup in the empty namespace must never be treated as an error. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.4 alwaysShowPlaceholderText (Use Custom XML Element Names as Default Placeholder Text)

This element specifies that each custom XML element specified using the customXml element within this document shall always show some form of in-document placeholder text presentation when it contains no run content. If the placeholder element (§17.5.2.25) is present in the custom XML element's properties, then this is the placeholder text displayed and this effect has no effect. If the placeholder element is omitted, then the application shall use the name of the element to generate default placeholder text in its place.

If this element is omitted, then custom XML markup which does not contain a placeholder element within its properties shall not display any placeholder text.

[*Example*: Consider the following WordprocessingML fragment from the document settings:

```
<w:alwaysShowPlaceholderText w:val="true" />
```

The alwaysShowPlaceholderText element has a value of `true`, which specifies that placeholder text must be generated using the element's name if no placeholder text is present. If two custom XML elements are defined as follows:

```
<w:customXml w:name="spec" … >
  <w:customXmlPr>
    <w:placeholder w:val="Type the name of the specification." />
  </w:customXmlPr>
</w:customXml>
…
<w:customXml w:name="spec" … >
</w:customXml>
```

The first custom XML element has placeholder text, and the second doesn't, so if this element is omitted, these two elements might be displayed as follows:

Type the name of the specification.

Notice that the second custom XML element has no placeholder text, and therefore is not displayed. However, when this element is present, then the application should generate default placeholder text in its place:

Type the name of the specification.

[spec]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The application generated default placeholder text from the element name, resulting in a value of `[spec]` in the document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.5 attachedSchema (Attached Custom XML Schema)

This element specifies that the custom XML schema whose target namespace matches the value specified in the val attribute should be associated with this document when it is loaded, if such a schema is available to the hosting application. Applications can also load and utilize any additional schemas as well as those explicitly mentioned here. [*Note*: These custom XML schemas can then be used to validate the structure of the custom XML markup in the document, etc. *end note*]

If no elements of this type are present, then no custom XML schemas have been explicitly associated with the contents of this document.

[*Example*: Consider the following WordprocessingML fragment from the document settings:

```
<w:attachedSchema w:val="http://www.example.com/schema1" />
<w:attachedSchema w:val="http://www.example.com/schema2" />
```

The attachedSchema elements specify that two custom XML schemas with namespaces of `http://www.example.com/schema1` and `http://www.example.com/schema2` should be associated with the custom XML markup in the current document. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.1.6 attachedTemplate (Attached Document Template)

This element specifies the location of a document template which shall be attached to the current WordprocessingML document if it is accessible and of a format supported by an application. Specifically, this element's `val` attribute shall contain the file path of the associated document template.

If this element is omitted, then the document shall not have an attached document template, and applications should use their default template in its place.

[*Example*: Consider a WordprocessingML document which is attached to a WordprocessingML template located on the local C drive within a file whose name is `c:\template.dotx`. This association is specified using the following WordprocessingML:

```
<w:attachedTemplate r:id="rId1" />
```

The attachedTemplate element contains the ID of the relationship which references the associated template. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

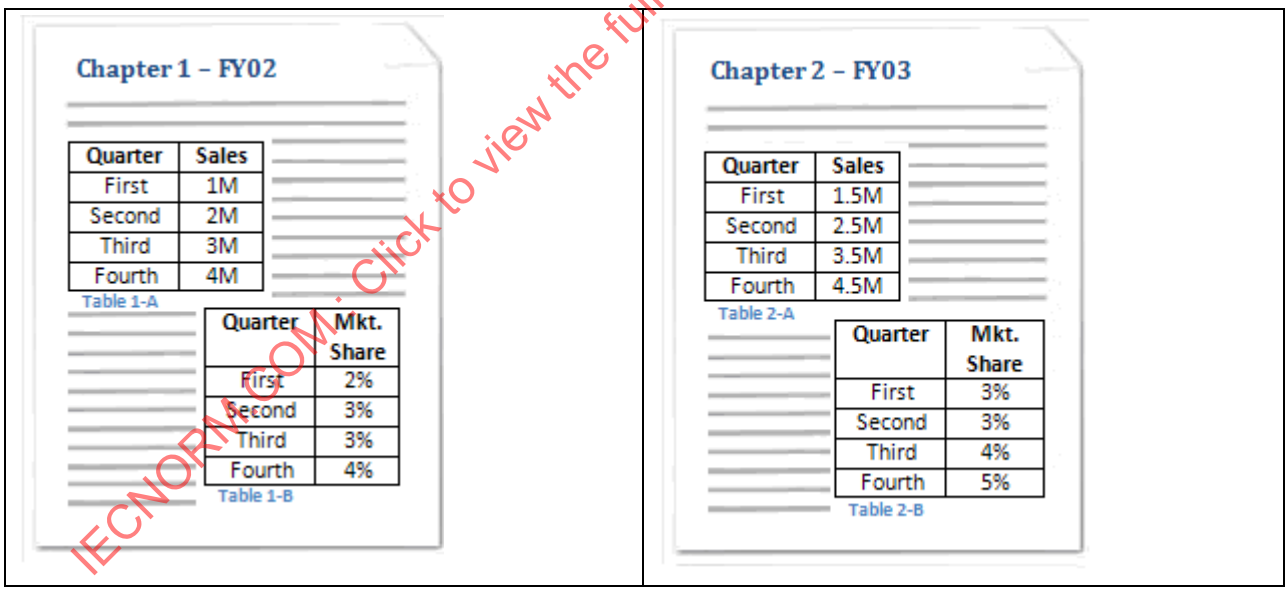
Attributes	Description
	[*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.15.1.7 autoCaption (Single Automatic Captioning Setting)

This element specifies what type(s) of objects shall automatically labelled with captions (§17.15.1.17), and with which captions the specified objects shall be labelled as defined in the caption element (§17.15.1.16).

[*Example*: Consider the diagram below illustrating a two page WordprocessingML document that has leveraged WordprocessingML to automatically label WordprocessingML tables with a specified caption when tables are inserted into the given document.



This type of automatic captioning is specified using the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:captions>
  <w:caption w:name="Table" w:pos="below" w:chapNum="1" w:heading="2"
w:numFmt="upperLetter" w:sep="hyphen" />
  <w:autoCaptions>
    <w:autoCaption w:name="wfwTable" w:caption="Table" />
  </w:autoCaptions>
</w:captions>
```

Here, the autoCaption element specifies through the name attribute being set equal to `wfwTable` that tables are automatically be labeled with the caption specified in the caption element whose name attribute is equal to `Table`, as the caption element's caption attribute has a value of `Table`. *end example*]

Attributes	Description

caption (Caption Used for Automatic Captioning)	Specifies the caption defined in using the caption element (§17.15.1.16) which shall be used to automatically label a given type of object inserted in a WordprocessingML document. The caption settings are linked by matching the value of this attribute with the name attribute of the corresponding caption element.
	
	[*Example*: Consider the WordprocessingML below
	
	`<w:captions>`
	`<w:caption w:name="table" w:pos="below" w:chapNum="1"`
	`w:heading="0" w:noLabel="1" w:numFmt="upperRoman" />`
	`<w:autoCaptions>`
	`<w:autoCaption w:name="Paint.Picture"`
	`w:caption="table" />`
	`</w:autoCaptions>`
	`</w:captions>`
	
	The autoCaption element specifies through the name attribute being set equal to `wfwTable` that tables are automatically be labeled with the caption whose name attribute is equal to `Table` (specified by the caption element's attribute name having a value of `Table`). *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

name (Identifier of Object to be Automatically Captioned)	Specifies a unique identifier which can be used to associate objects inserted into the document which are to be automatically labeled with a caption when inserted into the WordprocessingML document. [*Example*: Consider the WordprocessingML below specifying that WordprocessingML tables should be labeled with the `custom` caption: `<w:autoCaption w:name="wfwTables" w:caption="custom" />` The name attribute value of `wfwTables` specifies that WordprocessingML tables must be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

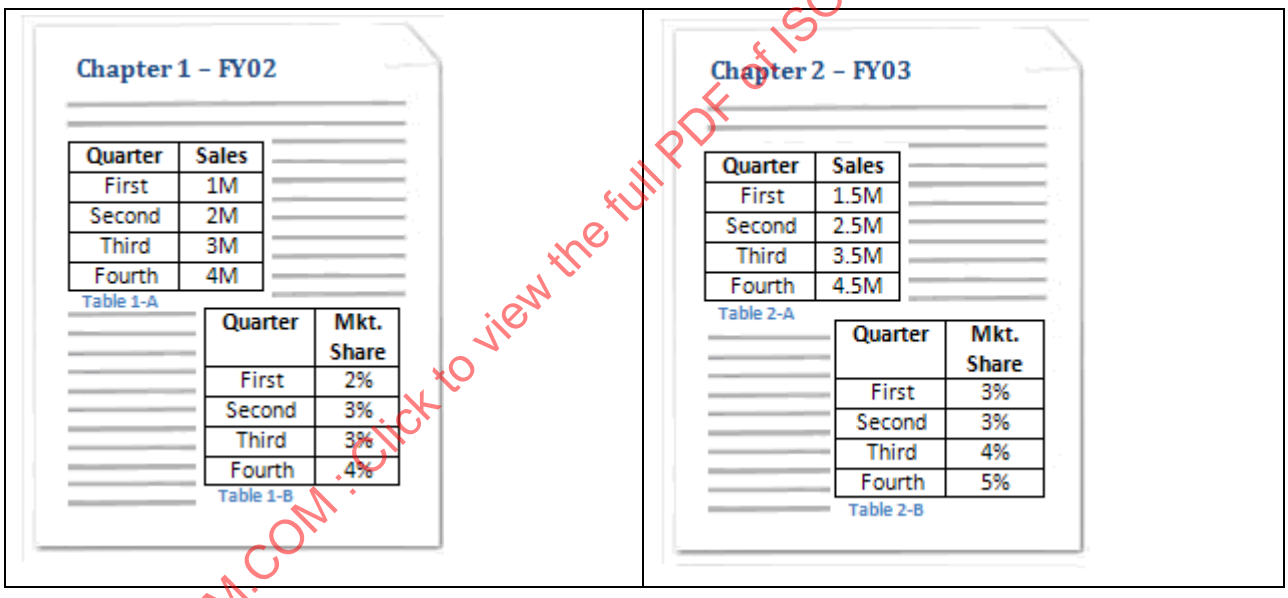
Attributes	Description
	labeled with the `custom` caption. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AutoCaption) is located in §A.1. *end note*]

17.15.1.8 autoCaptions (Automatic Captioning Settings)

This element specifies that one or more types of objects, when inserted into a WordprocessingML document, are automatically be labeled with a specific caption defined using the caption element (§17.15.1.16).

[*Example*: Consider the following example illustrating a two page WordprocessingML document that has leveraged WordprocessingML to automatically label WordprocessingML tables with a specified caption.



This type of automatic captioning is specified using the following WordprocessingML fragment:

```
<w:captions>
  <w:caption w:name="Table" w:pos="below" w:chapNum="1" w:heading="2"
w:numFmt="upperLetter" w:sep="hyphen" />
  <w:autoCaptions>
    <w:autoCaption w:name="wfwTable" w:caption="Table" />
  </w:autoCaptions>
</w:captions>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The autoCaptions element specifies set of objects that when inserted into a WordprocessingML document are automatically be labeled with a given caption. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AutoCaptions) is located in §A.1. *end note*]

17.15.1.9 autoFormatOverride (Allow Automatic Formatting to Override Formatting Protection Settings)

This element specifies whether formatting automatically applied by an application (i.e. not explicitly applied by a use or an application) shall be allowed to override formatting protection enabled via the formatting attribute on the documentProtection element (§17.15.1.9) when those formatting operations would add formatting which has been explicitly disabled. [*Example*: Automatically adding superscript to the `st` in the string `1st`. *end example*]

If this element is omitted, then no automatic formatting rule(s) shall be allowed to override the formatting restrictions enabled for the document.

[*Example* Consider a WordprocessingML document which has been protected such that a user must not be able to directly format text within the document. Consider also that the hosting application has been constructed such that if a user enters a ampersand, then one or more alphabetical characters, then another ampersand, that the alphabetical characters are to take on italicized formatting.

If the autoFormatOverride element is omitted or set to `false` and document protection is enabled, the aforementioned series of events does not cause the English alphabetical characters to be italicized as the document protection preventing formatting of the document in question supersedes the formatting to take place after these events. If this operation should not be prevented when active formatting restrictions are used, this would be specified using the following WordprocessingML:

```
<w:autoFormatOverride w:val="true"/>
```

The autoFormatOverride element's val attribute is equal to `true` specifying that the automatic formatting behavior shall be applied regardless of the formatting restrictions in place. *End Example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.10 autoHyphenation (Automatically Hyphenate Document Contents When Displayed)

This element specifies whether the content of a given WordprocessingML document should automatically be hyphenated by the hosting application before it is displayed, if the application supports this functionality.

If this element is omitted, then hyphenation shall not automatically be performed by application displaying this document.

[*Example*: Consider the images below illustrating a paragraph of text in a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text.

If the content in this document must automatically be hyphenated when it is displayed, that requirement would be specified using the following WordprocessingML in the document settings:

```
<w:autoHyphenation w:val="true" />
```

The resulting output might look like the following (depending on the application's hyphenation algorithm and the hyphenation zone setting (§17.15.1.53):

This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sample text. This is sam-ple text. This is sample text.

The autoHypehenation element has its val attribute equal to `true`, the document is automatically hyphenated and the word `sample`, beginning at the end of the second line, is hyphenated automatically and thus carried over onto the third line. Conversely, when the autoHypehnation element has its val attribute equal to `off`, the entire word `sample` is carried over to the third line as it was not hyphenated automatically and could not fit onto the second line. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.11 bookFoldPrinting (Book Fold Printing)

This element specifies if the contents of a given WordprocessingML document should be printed as signatures. *Signatures* are printed *sheets*, which depict several pages of a document that are folded and bound with other signatures to form a booklet, a set of which can be bound together to form a book like publication. Specifically, this element specifies that each page in a given WordprocessingML document should be oriented in a landscape fashion, divided in half vertically with two left margins emanating from the bisector of the page, and two right margins instantiated at the left and right side of each page.

This element is used in conjunction with the bookFoldPrintingSheets element (§17.15.1.12) to enable a WordprocessingML document to be printed such that the series of signatures printed can be folded and bound to create a booklet.

This element has no impact on the settings of printer leveraged by the hosting application. In other words, if the printer leveraged by the hosting application has been configured to print on one side of a page, including the WordprocessingML for this element has no effect.

If this element is omitted, then pages shall not be printed as signatures. If the bookFoldRevPrinting element (§17.15.1.13) is also specified, then this element shall be ignored.

[*Example*: Consider a four page WordprocessingML document with a 2,160 twentieths of a point (one and a half inch) left margin, and 1,440 twentieths of a point (one inch) bottom, right, and top margins using the pgMar

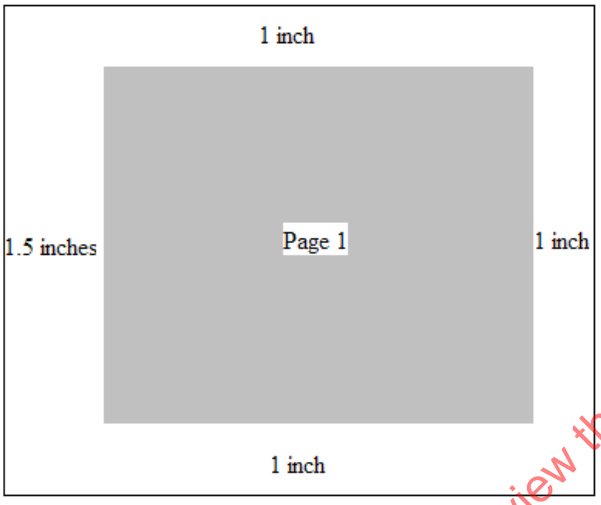
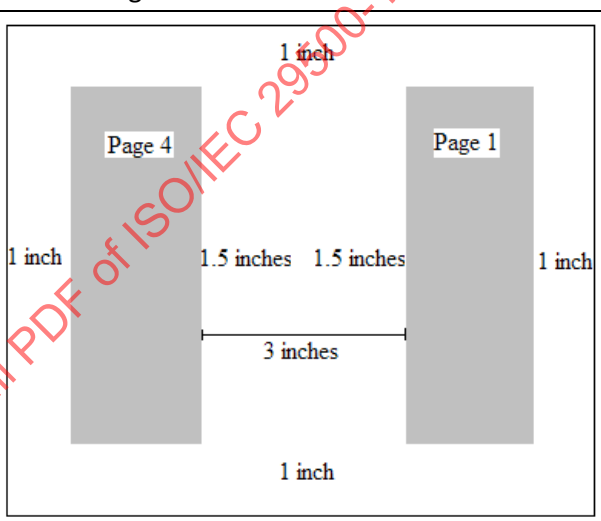
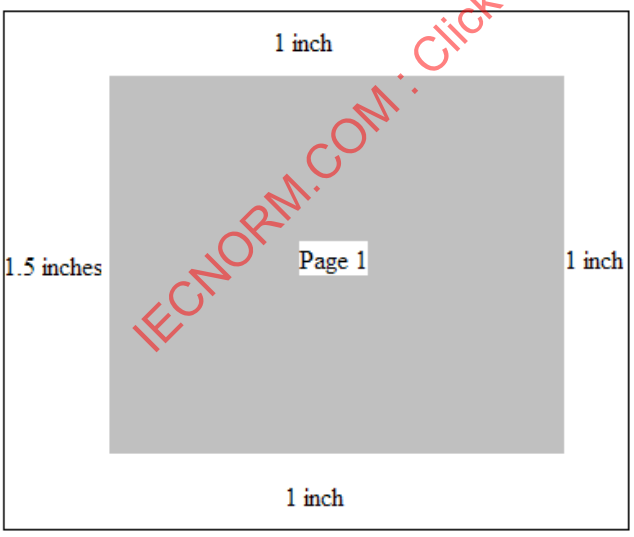
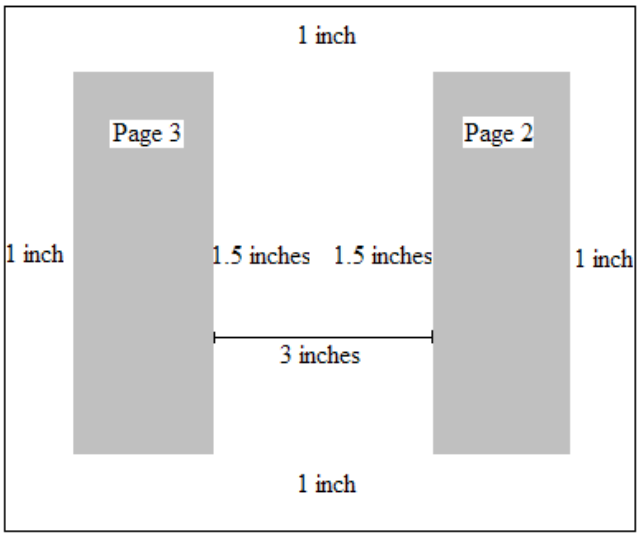
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

element (§17.6.11) surrounding the text extents of the page (represented by the gray shaded area in diagrams below). These page margins are specified using the following WordprocessingML:

```
<w:pgMar w:header="0" w:top="1440" w:right="1440" w:bottom="1440" w:left="2160"
  w:footer="720" w:gutter="0" />
```

The necessary WordprocessingML and consequential effect of setting the bookFoldPrinting element's val attribute to `true` versus `false` and the bookFoldPrintingSheets element's val attribute to `4`, is depicted graphically below—diagrams not drawn to scale:

<table>
<tr><td><code><w: bookFoldPrinting w:val="false"/></code></td><td><code><w: bookFoldPrinting w:val="true" /></code></td></tr>
<tr><td>First Printed Sheet</td><td>First Printed Signature</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Second Printed Sheet</td><td>Second Printed Signature</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Assuming the page was already oriented in a landscape fashion, setting the bookFoldPrinting element's val attribute to true divided the page in half vertically, with two left margins emanating from the bisector of the page, and right margins instantiated at the left and right side of each page, enabling two signatures to be printed.

In addition, this element is used in conjunction with the bookFoldPrintingSheets element to enable the given WordprocessingML document to be printed such that the series of signatures printed can be folded and bound to create a booklet. Specifically, the signatures can be placed back to back, with top the bottom of each sheet aligned, and folded such that a booklet is created. *end example*]

[*Note*: This element could also be leveraged by the hosting application to notify the application to display two pages per sheets within its user interface to allow for a WYSIWYG user experience. *end note*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.12 bookFoldPrintingSheets (Number of Pages Per Booklet)

This element shall be used in conjunction with the bookFoldPrinting (§17.15.1.11) and bookFoldRevPrinting (§17.15.1.13) elements to specify the number of pages to be included in each booklet when printing a series of signatures. Signatures are printed *sheets*, which depict several pages of a document that are to be folded and bound with other signatures to form a booklet. Booklets can be bound together to form a book like publication.

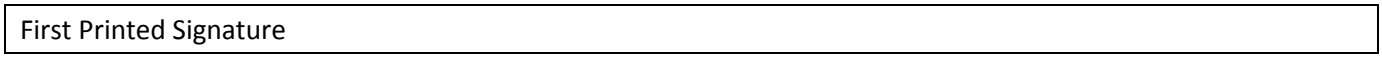
If this element is omitted, then its default behavior shall be to print the contents of the content on a single sheet. A *sheet* is a single piece of paper which is folded and cut to produce a book.

[*Example*: Consider a four page WordprocessingML document is printed as a set of two signatures to be compiled into a single booklet. This setting would be specified using the following WordprocessingML fragment in the document settings part:

```
<w:bookFoldPrinting w:val="true" />
<w:bookFoldPrintingSheets w:val="4" />
```

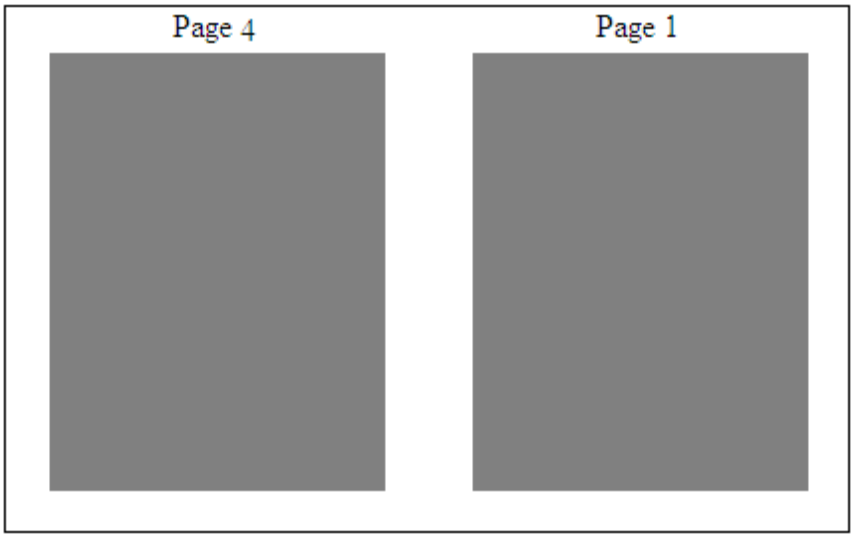
The bookFoldPrintingSheets element's val attribute specifies that 4 pages must be included in each booklet. Since each signature contains two pages and are printed such that the signatures can be placed back to back, with top the bottom of each sheet aligned, and folded such that the booklet is created, a booklet containing four pages distributed over two signatures can be created.

This setting is depicted visually using the illustration below (gray shading represents a page):

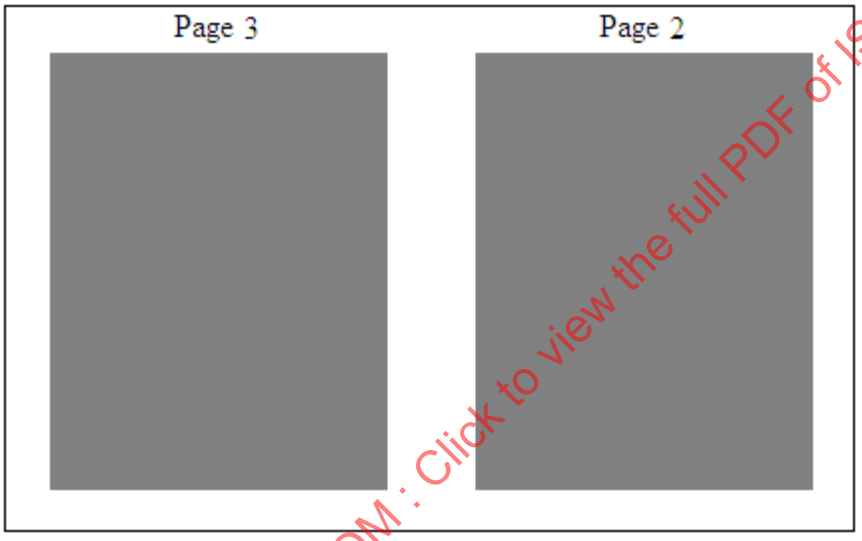


IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



Second Printed Signature



end example]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.15.1.13 bookFoldRevPrinting (Reverse Book Fold Printing)

This element specifies if pages of a given WordprocessingML document are to be printed as signatures in reverse order. *Signatures* are printed *sheets*, which depict several pages of a document that are folded and bound with other signatures to form a booklet, a set of which can be bound together to form a book like publication. Specifically, this element specifies that each page in a given WordprocessingML document should be oriented in a landscape fashion and divided in half vertically, with two left margins emanating from the bisector of the page, and right margins instantiated at the left and right side of each page.

In addition, this element is used in conjunction with the bookFoldPrintingSheets element (§17.15.1.12) to enable given WordprocessingML document to be printed such that the series of signatures printed can be folded and bound to create a booklet.

This element has no impact on the settings of printer leveraged by the hosting application. In other words, if the printer leveraged by the hosting application has been configured to print on one side of a page, including the WordprocessingML for this element has no effect.

If this element is omitted, then pages shall not be printed as reverse book fold signatures. If the bookFoldPrinting element (§17.15.1.11) is also specified, then that element shall be ignored, and this element shall be used instead.

[*Example*: Consider a four page WordprocessingML document with a 2,160 twentieths of a point (one and a half inch) left margin, and 1,440 twentieths of a point (one inch) bottom, right, and top margins using the pgMar element (§17.6.11) surrounding the text extents of the page (represented by the gray shaded area in diagrams below). These page margins are specified using the following WordprocessingML:

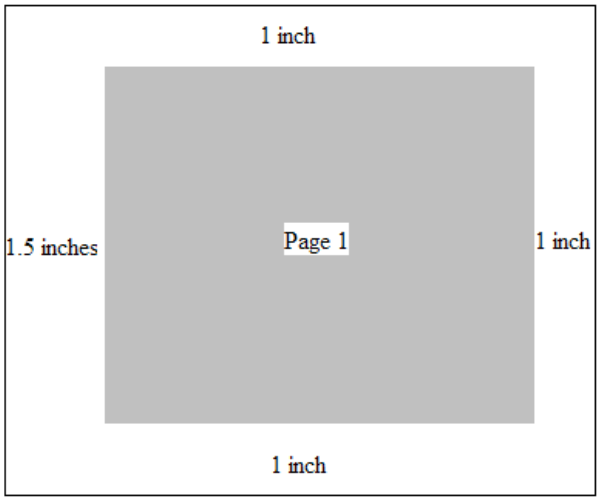
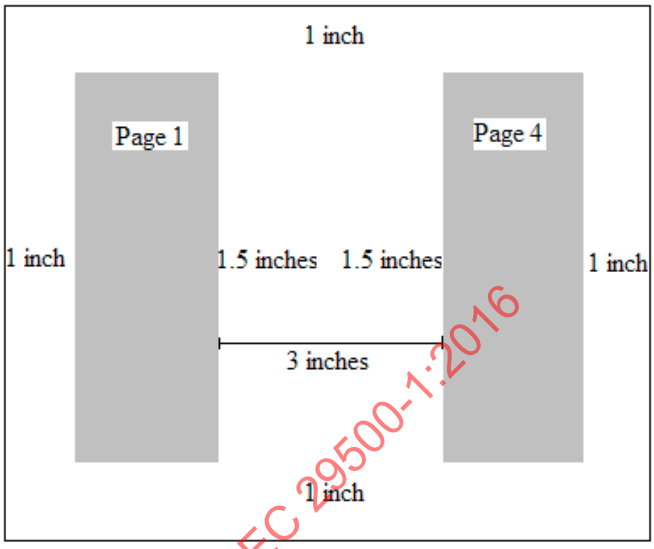
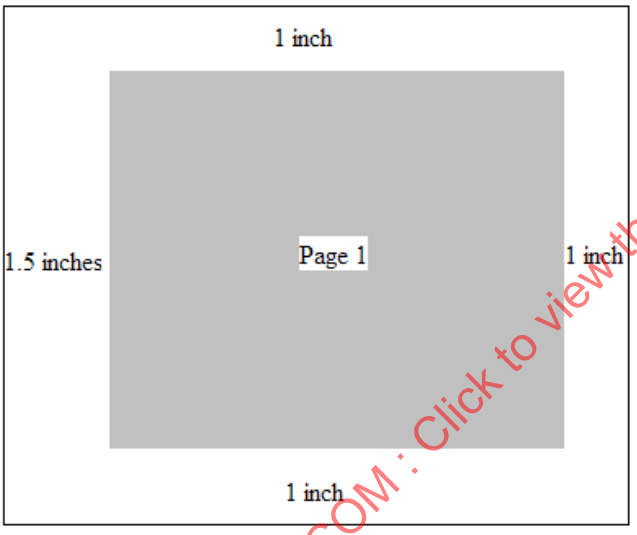
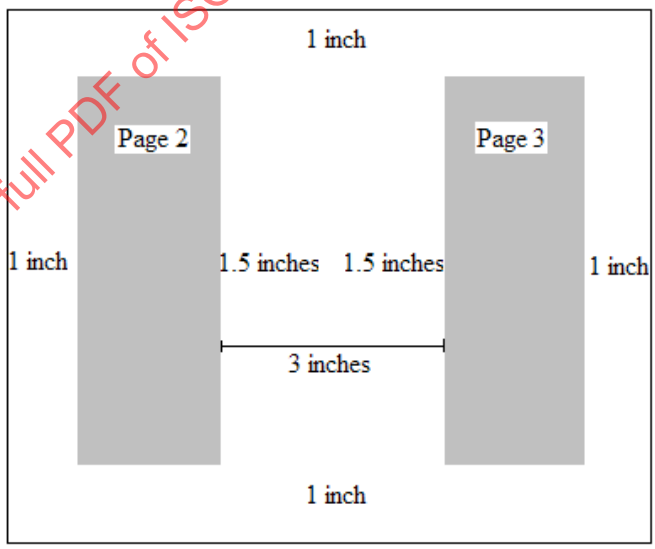
```
<w:pgMar w:header="0" w:top="1440" w:right="1440" w:bottom="1440" w:left="2160"
  w:footer="720" w:gutter="0" />
```

The necessary WordprocessingML and consequential effect of setting the bookFoldRevPrinting element's val attribute to `true` versus `false` and the bookFoldPrintingSheets element's val attribute to `4`, is depicted graphically below—diagrams not drawn to scale:

`<w: bookFoldRevPrinting w:val="false"/>`	`<w: bookFoldRevPrinting w:val="true"/>` `<w: bookFoldPrintingSheets w:val="4"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><td>First Printed Sheet</td><td>First Printed Signature</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Second Printed Sheet</td><td>Second Printed Signature</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Assuming the page was already oriented in a landscape fashion, setting the bookFoldRevPrinting element's val attribute to true divided the page in half vertically, with two left margins emanating from the bisector of the page, and right margins instantiated at the left and right side of each page, enabling two signatures to be printed.

In addition, this element is used in conjunction with the bookFoldPrintingSheets element to enable the given WordprocessingML document to be printed such that the series of signatures printed can be folded and bound to create a booklet. Specifically, the signatures can be placed back to back, with top the bottom of each sheet aligned, and folded such that a booklet is created. *end example*]

©ISO/IEC 2016 – All rights reserved

[*Note*: This element could also be leveraged by the hosting application to notify the application to display two pages per sheets within its user interface to allow for a WYSIWYG user experience. *end note*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.14 bordersDoNotSurroundFooter (Page Border Excludes Footer)

This element specifies that a given WordprocessingML document's page border specified using the pgBorders element (§17.6.10) should not surround contents of the footer.

If this element is omitted, then the page border shall not exclude the footer on the page. As well, this element shall be ignored if the pgBorders element has an offsetFrom attribute which is not equal to `text`.

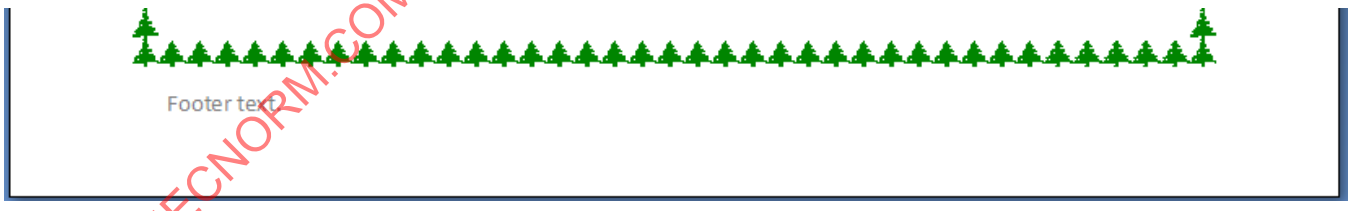
[*Note*: If the pgBorders element has a offsetFrom attribute equal to `page`, the bordersDoNotSurroundFooter element is ignored as specifying the pgBorders element with a offsetFrom attribute equal to page is to specify that the positioning of borders within the document shall be calculated relative to the edge of the page and therefore irrespective of document content in the footer. *end note*]

[*Example*: Consider the following page in a WordprocessingML document:

If this WordprocessingML document is modified to leverage the behavior enabled by this element, this setting would be specified using the following WordprocessingML fragment in the document settings:

```
<w:bordersDoNotSurroundFooter w:val="true"/>
```

The bordersDoNotSurroundFooter element's val attribute is equal to `true` specifying that the page border shall not surround the text extents of the footer, as follows:



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.15 bordersDoNotSurroundHeader (Page Border Excludes Header)

This element specifies that a given WordprocessingML document's page border specified using the pgBorders element (§17.6.10) should not surround contents of the header.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then the page border shall not exclude the header on the page. As well, this element shall be ignored if the pgBorders element has a offsetFrom attribute which is not equal to `text`.

[*Note*: If the pgBorders element has a offsetFrom attribute equal to `page`, the bordersDoNotSurroundHeader element is ignored as specifying the pgBorders element with a offsetFrom attribute equal to page is to specify that the positioning of borders within the document shall be calculated relative to the edge of the page and therefore irrespective of document content in the header. *end note*]

[*Example*: Consider the following page in a WordprocessingML document:

If this WordprocessingML document is modified to leverage the behavior enabled by this element, this setting would be specified using the following WordprocessingML fragment in the document settings:

```
<w:bordersDoNotSurroundHeader w:val="true"/>
```

The bordersDoNotSurroundHeader element's val attribute is equal to `true` specifying that the page border shall not surround the text extents of the header, as follows:

end example]

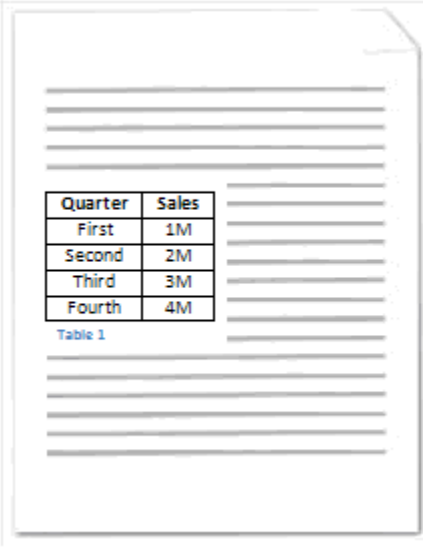
This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.16 caption (Single Caption Type Definition)

This element specifies the contents and positioning for captions which can be used to automatically label objects in a WordprocessingML document. A *caption* is a string that labels an object included in a WordprocessingML document, and typically consists of a string plus a field which numbers this item within a collection of similar objects.

[*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a caption:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



In this diagram, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of the string `Table` followed by a field whose result is a decimal number. The settings which automatically produced this form of caption are specified using the following WordprocessingML fragment:

```
<w:captions>
  <w:caption w:name="Table" w:pos="below" w:numFmt="decimal" />
</w:captions>
```

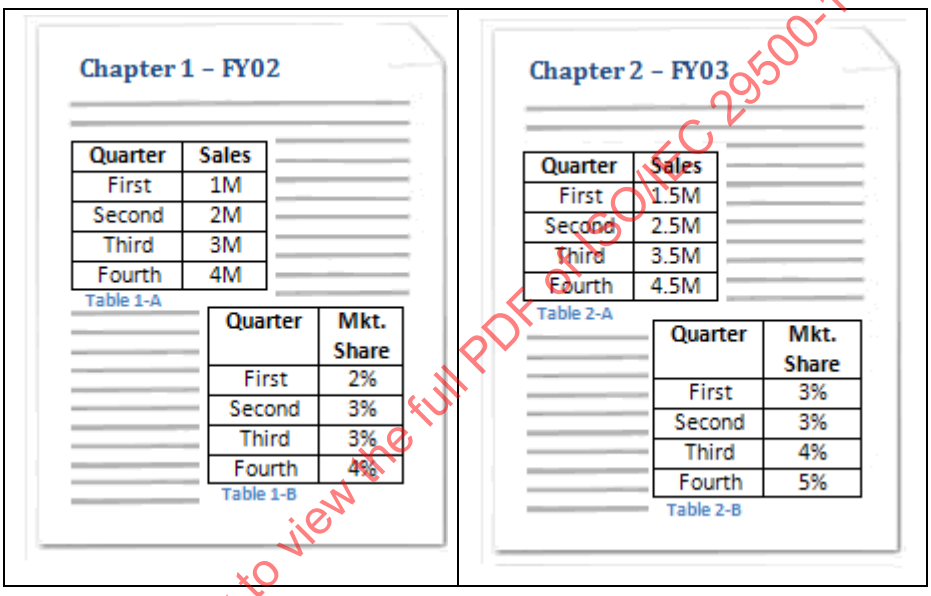
The caption element specifies the parameters for the resulting caption to be used to automatically label content within the WordprocessingML document. Specifically, the name and numFmt attributes specify that captions of this caption type inserted in the given WordprocessingML document shall consist of the string `Table` followed by an incrementing decimal number field. In addition, the pos attribute specifies that these captions shall be placed below the object they are used to label.

WordprocessingML is designed such that the caption element can be used in conjunction with applications to provide a dynamic captioning experience. In other words, an application can use the WordprocessingML in the example above to automatically insert a caption consisting of the string `Table` followed by an incrementing decimal number field below tables when tables are inserted into a WordprocessingML document as defined by the autoCaption element (§17.15.1.7). *end example*]

Attributes	Description
chapNum (Include Chapter Number in Field for Caption)	Specifies whether or not to display numbering associated with the most recent chapter heading in the WordprocessingML document within the caption field. A *chapter heading* is a paragraph of text within a WordprocessingML document that is formatted with a style that has been specified by the heading attribute to demarcate chapters in documents. Only a style with its styleID attribute equal to `Heading1`, `Heading2`, `Heading3`,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	`Heading4`, `Heading5`, `Heading6`, `Heading7`, `Heading8`, or `Heading9` can be specified as the style used to demarcate chapters in a document. The choice of which of these heading levels shall be used to determine the current chapter number is defined by the value of the corresponding heading attribute. [*Example*: `Heading1` is used as the chapter heading when chapNum is `true` and heading is `1`. *end example*] If this attribute is omitted, then chapter numbers shall not be included in the resulting caption. [*Example*: Consider the diagram below:  This diagram depicts a WordprocessingML document containing two chapters, each containing two tables labeled with captions. The Heading 2 style has been associated with chapter headings and applied to the strings: `Chapter 1 - FY02` and `Chapter 2 - FY03` in this document. Specifically, the style used to demarcate chapters, is the style with a styleID attribute equal to `Heading2` as specified by the heading attribute value of `2` in the WordprocessingML for this caption, defined as follows: `<w:caption w:name="Table" w:pos="below" w:chapNum="true" w:heading="2" w:numFmt="upperLetter" w:sep="hyphen" />` The chapNum attribute has a value of `true`, specifying that the captions used to label the tables within this document contains a symbol corresponding to the one-based index of the chapter in which it is contained. This can be seen in that the captions in Chapter 1 contain a `1`, while the captions in Chapter 2 contain a `2`, each corresponding with their respective chapter number. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

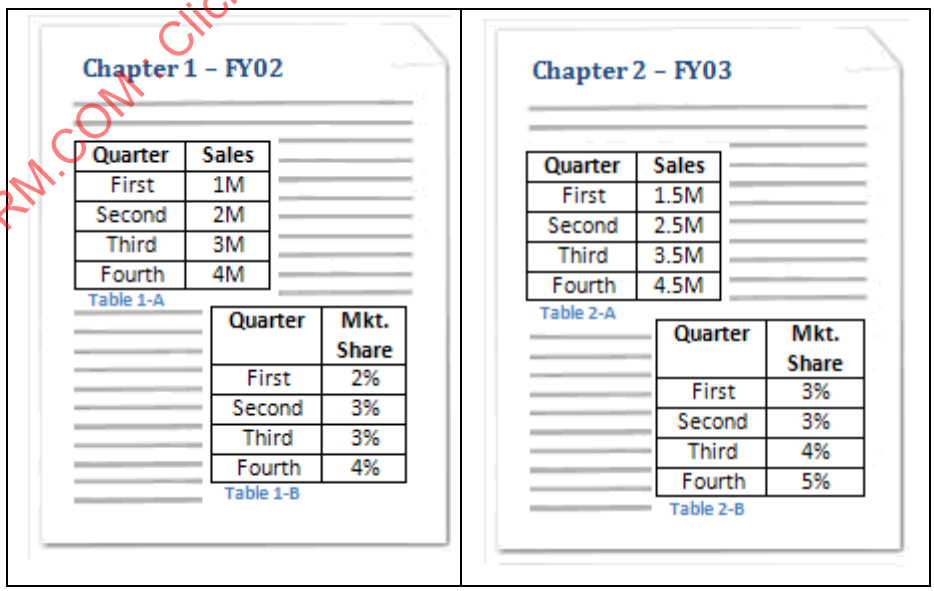
©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>example]

The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).</td></tr>
<tr><td>heading (Style for Chapter Headings)</td><td>Specifies the given style that is used to demarcate chapter headings in a document.

This value is used to link the chapter headings with paragraphs with a styleID attribute as follows:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Style with styleID of Heading1</td></tr>
<tr><td>2</td><td>Style with styleID of Heading2</td></tr>
<tr><td>3</td><td>Style with styleID of Heading3</td></tr>
<tr><td>4</td><td>Style with styleID of Heading4</td></tr>
<tr><td>5</td><td>Style with styleID of Heading5</td></tr>
<tr><td>6</td><td>Style with styleID of Heading6</td></tr>
<tr><td>7</td><td>Style with styleID of Heading7</td></tr>
<tr><td>8</td><td>Style with styleID of Heading8</td></tr>
<tr><td>9</td><td>Style with styleID of Heading9</td></tr>
<tr><td>Any other value</td><td>Application-defined. Can be ignored.</td></tr>
</table>
If this attribute is omitted, then its value shall be assumed to be 1.

[Example: Consider the diagram below:



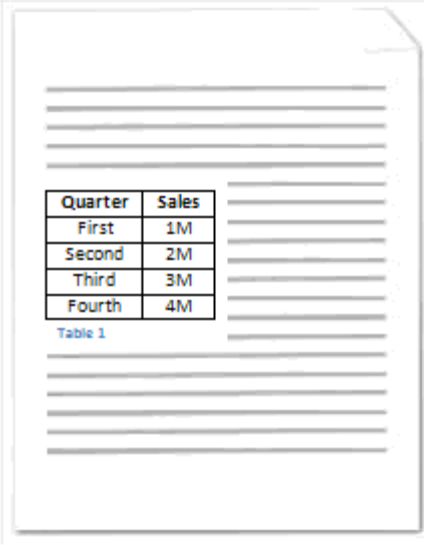
This diagram depicts a WordprocessingML document containing two chapters, each</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

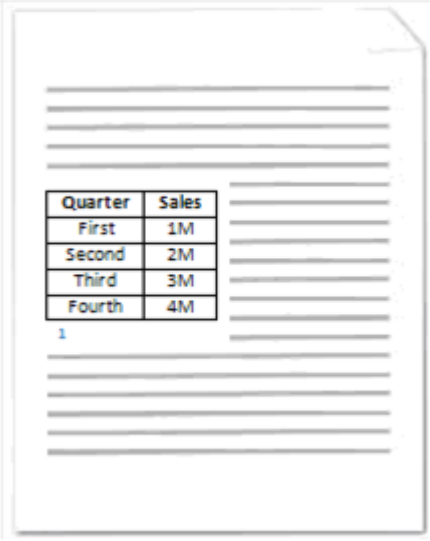
Attributes	Description
	containing two tables labeled with captions. The Heading 2 style has been associated with chapter headings and applied to the strings: `Chapter 1 - FY02` and `Chapter 2 - FY03` in this document. Specifically, the style used to demarcate chapter headings is the style with its styleID attribute equal to `Heading`" as specified by the heading attribute value of 2 in the WordprocessingML below. `<w:caption w:name="Table" w:pos="below" w:chapNum="1" w:heading="2" w:numFmt="upperLetter" w:sep="hyphen" />` In other words, the WordprocessingML above can be used to label tables inserted in a given WordprocessingML document generated by an application with a caption consisting of: the string `Table` followed by a decimal number corresponding with the chapter number in which the table is present, a hyphen as defined in the sep attribute, and a capital English letter defined by the numFmt attribute corresponding with the given table's ordering within the current chapter. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).
name (Caption Type Name)	Specifies the literal string component of this caption. This value is used as follows: • It is added to the field containing the chapter number and item number of this object when a caption is automatically added to the document. • It is used to uniquely label this caption type, allowing it to be linked with classes of objects via the autoCaption element (§17.15.1.7) • It can be used to label this caption type in a user interface. [*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a caption:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	 In this diagram, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of the string `Table` followed by a decimal number. This caption format is specified with the following WordprocessingML: `<w:caption w:name="Table" w:pos="below" w:numFmt="decimal" />` Specifically, the name attribute specifies that the first part of the string that comprises the give caption shall consist of the string `Table`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
noLabel (Do Not Include Name In Caption)	Specifies if the string specified in the name attribute shall be included in the resulting caption when it is automatically added to the document. If set to true, then the label text in the name attribute is omitted when adding the caption. If this attribute is omitted, then the name shall be added to the caption. [*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a caption:

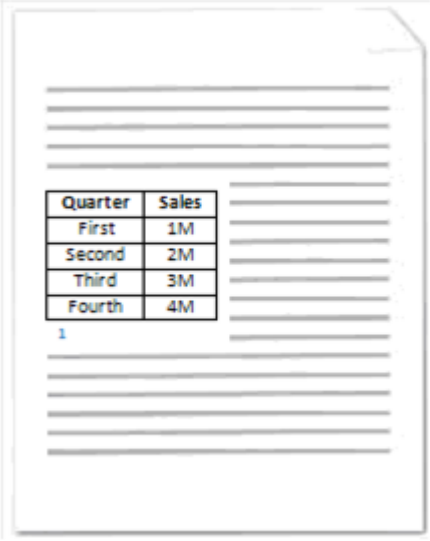
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 In this diagram, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of only a decimal number. This caption format is specified using the following WordprocessingML: `<w:caption w:name="Custom" w:pos="below" w:noLabel="true" w:numFmt="decimal" />` Here, the noLabel attribute is equal to `true` specifying that when this caption format is automatically added, it must not include the label. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
numFmt (Caption Numbering Format)	Specifies the format of the numbering which shall be included in an automatically generated caption to specify the index of this item in that collection (within the current chapter if chapNum is specified, or within the current document story). If this attribute is omitted, then its default value shall be assumed to be `decimal`. [*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a caption:

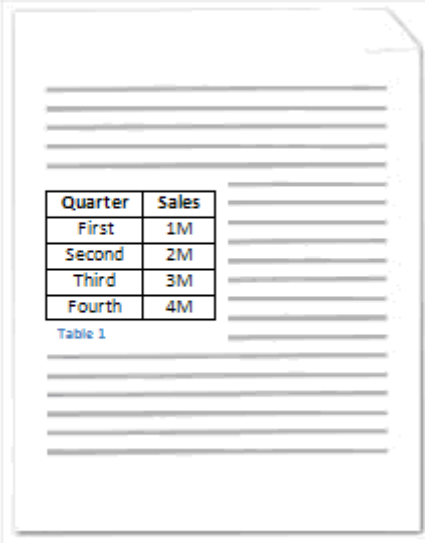
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 In this example, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of only a decimal number. This caption format is specified using the following WordprocessingML: `<w:caption w:name="Custom" w:pos="below" w:noLabel="true" w:numFmt="decimal" />` Here, the numFmt attribute is equal to `decimal`, specifying that a decimal number shall be included in the table caption when it is automatically inserted. *End Example*] The possible values for this attribute are defined by the ST_NumberFormat simple type (§17.18.59).
pos (Automatic Caption Placement)	Specifies how an automatically inserted caption shall be positioned relative to the object that it is captioning. If this attribute is omitted, then the default value shall be `below`. [*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a *caption*.

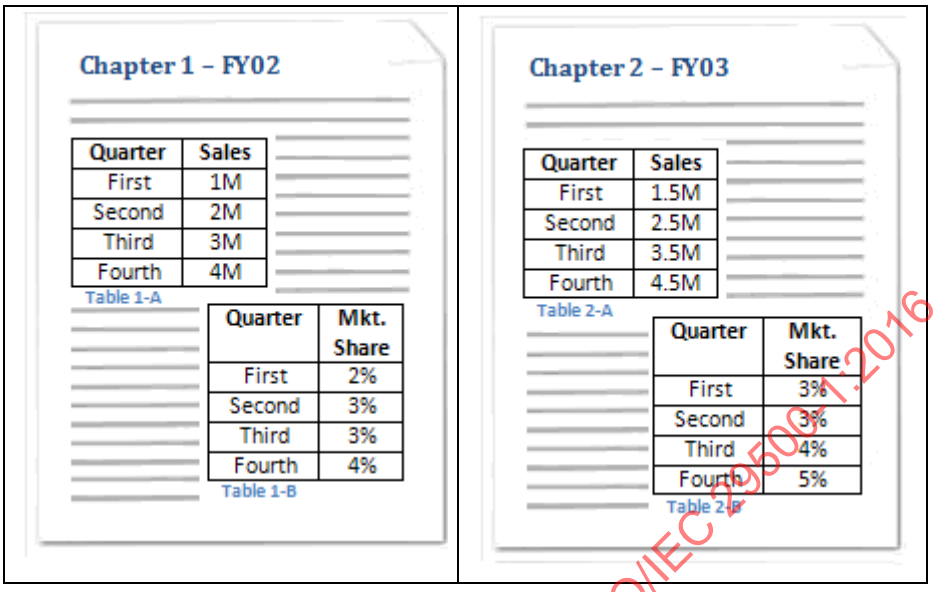
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 In this diagram, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of the string `Table` followed by a decimal number. This caption format is specified using the following WordprocessingML: `<w:caption w:name="Table" w:pos="below" w:numFmt="decimal" />` The pos attribute specifies that the given caption shall be placed `below` the object it is labelling. *end example*] The possible values for this attribute are defined by the ST_CaptionPos simple type (§17.18.5).
sep (Chapter Number/Item Index Separator)	Specifies the character which shall be used to separate the chapter number used in this caption from the caption item numbering. A caption format consists of three components: • The (optional) literal string • The (optional) chapter number • The index of this caption within the chapter/document When the latter two items are both present, they are delimited using the chapter separator specified by this attribute. If this attribute is omitted, then its default value shall be hyphen. If the chapter number is not part of the caption format, then this parameter shall be ignored. [*Example*: Consider the diagram below:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 This diagram depicts a WordprocessingML document containing two chapters, each containing two tables labeled with captions. The Heading 2 style has been associated with chapter headings and applied to the strings: `Chapter 1 - FY02` and `Chapter 2 - FY03` in this document. Specifically, the style used to demarcate chapter headings is the style with a styleID attribute equal to `Heading2` as specified by the heading attribute value of `2` in the WordprocessingML below. `<w:caption w:name="Table" w:pos="below" w:chapNum="1" w:heading="2" w:numFmt="upperLetter" w:sep="hyphen" />` The sep attribute value of hyphen specifies that the chapter number and caption index must be separated by a hyphen character when displayed in the document. *end example*] The possible values for this attribute are defined by the ST_ChapterSep simple type (§17.18.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Caption) is located in §A.1. *end note*]

17.15.1.17 captions (Caption Settings)

This element specifies the presence of information about captions in a given WordprocessingML document. This information is divided into two components:

- The child element caption defines the format for a single type of caption to be automatically added to the document.

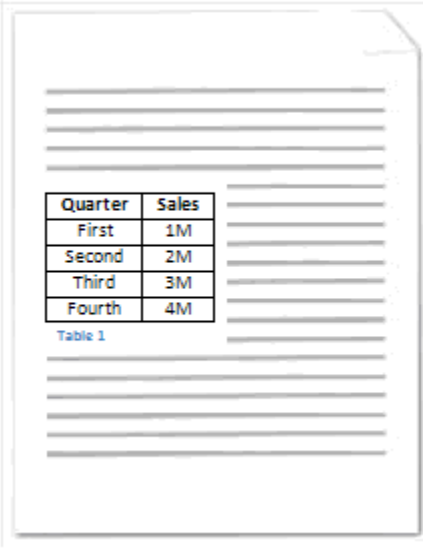
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- The child element autoCaptions defines the types of objects to which a caption format shall automatically be applied.

This information should be used to determine the captions which are automatically added to objects when they are inserted into a WordprocessingML document. [*Note*: This setting is typically ignored unless it is specified in an application's default template. *end note*]

[*Example*: Consider the diagram below illustrating a WordprocessingML document containing a table that has been labeled with a caption:



In this diagram, the table contained in the WordprocessingML document has been labeled by inserting a caption below the table consisting of the string Table followed by a decimal number. This automatically inserted caption format is specified using the following WordprocessingML:

```
<w:captions>
  <w:caption w:name="Table" w:pos="below" w:numFmt="decimal" />
</w:captions>
```

Here, the captions element specifies the presence of one or more caption formats in a given WordprocessingML document with its child element caption. Specifically, the child element caption specifies a single type of caption to be used within the WordprocessingML document. *end example*]

Captioning leverages fields (§17.16.5) to label objects with reference to either:

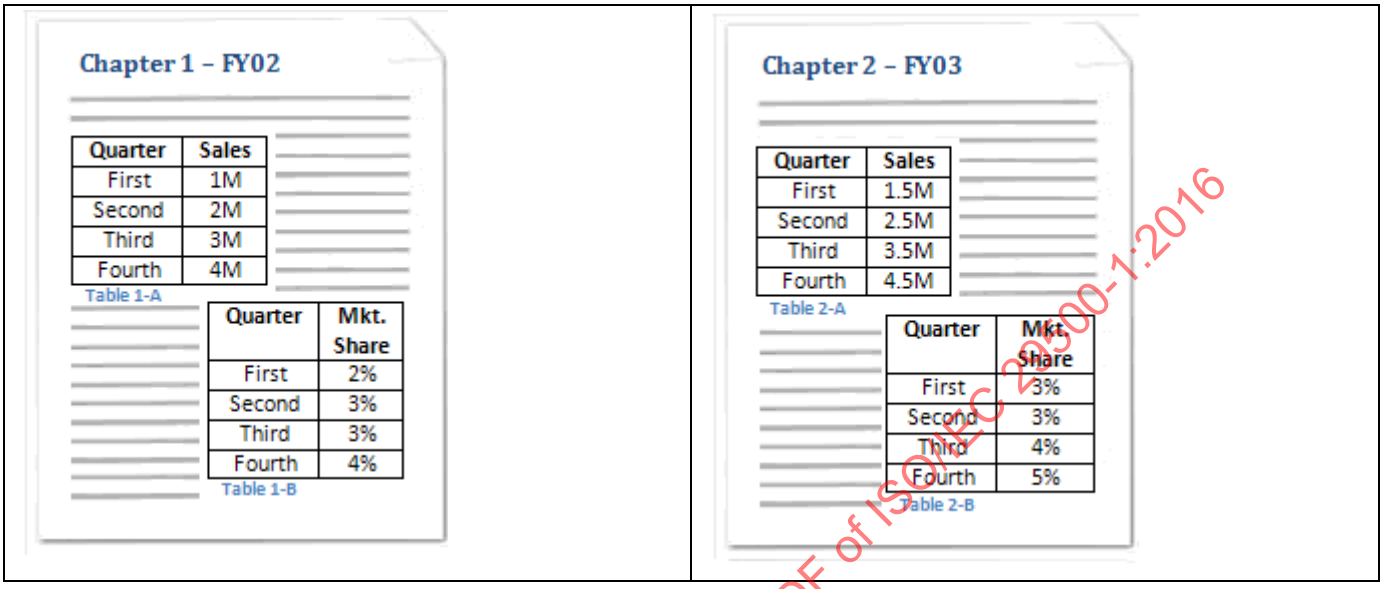
- Other captioned objects within a given document
- Other captioned objects within the same chapter in a given document (when chapter numbers are added by specifiying the chapNum attribute on the caption type.

  A *chapter* is a section of text within a WordprocessingML document that is preceded by content with a style that has been specified by to demarcate chapters in documents. Only one style can be specified as

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

the style used for a single caption type to demarcate chapters in a document. A chapter ends immediately above the next instance of content with the style used to demarcate chapters.

[*Example*: Consider the diagram below:



This diagram depicts a WordprocessingML document containing two chapters, each containing two tables labeled with captions. The style associated with chapter demarcation has been applied to the strings: `Chapter 1 - FY02` and `Chapter 2 - FY03` in this document. Specifically, the style used to demarcate chapters is the style with its styleID attribute equal to `Heading2` as specified by the heading attribute value of 2 in the WordprocessingML for the caption format:

```
<w:caption w:name="Table" w:pos="below" w:chapNum="1" w:heading="2"
w:numFmt="upperLetter" w:sep="hyphen" />
```

In other words, the WordprocessingML above can be used to label objects (in this case, tables) inserted in a given WordprocessingML document generated by an application with a caption consisting of: the string Table followed by a decimal number corresponding with the chapter number in which the table is present, a hyphen, and a capital English letter corresponding with the given table's index within the given chapter. *end example*]

[*Note*: WordprocessingML is designed such that the caption element can be used in conjunction with applications to provide a dynamic captioning experience. In other words, an application can use the WordprocessingML in the example above to automatically insert a caption consisting of the string `Table` followed by an incrementing decimal number field below tables when tables are inserted into a WordprocessingML document as defined by the autoCaption element (§17.15.1.7). *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Captions) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.18 characterSpacingControl (Character-Level Whitespace Compression)

This element specifies how full-width characters in the current WordprocessingML document should be compressed to remove additional whitespace when the contents of this document are displayed, specifically by specifying the set(s) of characters which can be compressed to remove additional whitespace. [*Note*: The behavior of this element is functionally identical to the CSS text-justify-trim property. *end note*]

If this element is omitted, then the default value shall be `dontCompress`.

[*Example*: Consider the WordprocessingML below:

```
<w:characterSpacingControl w:val="doNotCompress" />
```

The characterSpacingControl element has a val attribute value of `doNotCompress`, which specifies that no character compression shall be applied to any character when the document is displayed. *end example*]

Attributes	Description
val (Value)	Specifies the set(s) of characters which should be compressed when the contents of this document are displayed. [*Example*: Consider a WordprocessingML document for which only full-width punctuation characters must have their whitespace compression applied. This requirement would be specified using the following WordprocessingML: `<w:characterSpacingControl w:val="compressPunctuation"/>` The val attribute value of `compressPunctuation` specifies that character compression must be applied to full-width punctuation characters only when the document is displayed. *end example*] The possible values for this attribute are defined by the ST_CharacterSpacing simple type (§17.18.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CharacterSpacing) is located in §A.1. *end note*]

17.15.1.19 clickAndTypeStyle (Paragraph Style Applied to Automatically Generated Paragraphs)

This element specifies the paragraph style, specified using the style element, which shall be applied to paragraphs which are automatically created when text is inserted into a WordprocessingML document in an area of the document that has no other style associated with it. This style is referenced via the val attribute, which stores the style ID of the style (stored in the styleId attribute on the style definition).

[*Guidance*: Consider a WordprocessingML document opened in an application that allows users to place their cursor anywhere within the document editing canvas and enter text. The clickAndTypeStyle element should be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

used to specify the paragraph style to be associated with the paragraph of text entered after a user places their cursor somewhere in the blank document that results in the generation of new paragraphs. *end guidance*]

If this element is omitted, then the default paragraph style (the paragraph style whose default attribute is set to true), shall be used for automatically generated paragraphs. If the style whose styleId is specified using the val attribute is not a paragraph style or does not exist in the document, then the default paragraph style shall be used instead.

[*Example*: Consider a WordprocessingML document that has specified that paragraphs which are automatically created when text is inserted in a given area of the document which has no other style associated with it must be associated with the paragraph style that has a styleId equal to `BalloonText`.

This is accomplished by specifying a clickAndTypeStyle element with a val attribute equal to the value of the ID of the desired style. This constraint would be specified using the following WordprocessingML:

```
<w:clickAndTypeStyle w:val="BalloonText" />
```

The corresponding style in the styles part would be defined as follows:

```
<w:style w:type="paragraph" w:styleId="BalloonText">
  …
</w:style>
```

The clickAndTypeStyle element specifies the use of the paragraph style with the style ID of `BalloonText`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.1.20 clrSchemeMapping (Theme Color Mappings)

This element specifies the theme color, stored in the document's Theme part to which the value of this theme color shall be mapped. This mapping enables multiple theme colors to be chained together.

[*Example*: Consider a WordprocessingML document that must have the theme color value `background1` mapped to the theme color `light1` as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:clrSchemeMapping w:bg1="light1" />
```

The clrSchemeMapping element's attribute background1 has a value of `light1`, specifying that theme color value `background1` must be mapped to the theme color light1. *end example*]

Attributes	Description
accent1 (Accent 1 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent1` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color `accent1` mapped to the theme color lt1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent1="light1" />` The accent1 attribute has a value of `light1`, specifying that uses of the theme color value `accent1` must be mapped to the theme color lt1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
accent2 (Accent 2 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent2` theme color shall be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	[*Example*: Consider a WordprocessingML document that must have the references to the theme color `accent2` mapped to the theme color hlink as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent2="hyperlink" />` The accent2 attribute has a value of `hyperlink`, specifying that uses of the theme color value `accent2` must be mapped to the theme color hlink. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
accent3 (Accent3 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent3` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color `accent3` mapped to the theme color dk1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent3="dark1" />` The accent3 attribute has a value of `dark1`, specifying that uses of the theme color value `accent3` must be mapped to the theme color dk1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
accent4 (Accent4 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent4` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color `accent4` mapped to the theme color dk2 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent4="dark2" />` The accent4 attribute has a value of `dark2`, specifying that uses of the theme color value `accent3` must be mapped to the theme color dk2. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
accent5 (Accent5 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent5` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color `accent5` mapped to the theme color accent1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent5="accent1" />` The accent5 attribute has a value of `accent1`, specifying that uses of the theme color value `accent5` must be mapped to the theme color accent1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
accent6 (Accent6 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `accent6` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color `accent6` mapped to the theme color accent1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:accent6="accent1" />` The accent6 attribute has a value of `accent1`, specifying that uses of the theme color value `accent6` must be mapped to the theme color accent1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
bg1 (Background 1 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the `light1` theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color bg1 mapped to the theme color lt2 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:clrSchemeMapping w:bg1="light2" /> The bg1 attribute has a value of light2, specifying that uses of the theme color value bg1 must be mapped to the theme color lt2. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
bg2 (Background 2 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the light2 theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color bg2 mapped to the theme color dk1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: <w:clrSchemeMapping w:bg2="dark1" /> The bg2 attribute has a value of dark1, specifying that uses of the theme color value bg2 must be mapped to the theme color dk1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
followedHyperlink (Followed Hyperlink Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the followedHyperlink theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color followedHyperlink mapped to the theme color hyperlink as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: <w:clrSchemeMapping w:followedHyperlink="hyperlink" /> The followedHyperlink attribute has a value of hyperlink, specifying that uses of the theme color value followedHyperlink must be mapped to the theme color hyperlink. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
hyperlink (Hyperlink Theme	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Color Mapping)	If this attribute is omitted, then the hyperlink theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color hyperlink mapped to the theme color accent1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:hyperlink="accent1" />` The hyperlink attribute has a value of accent1, specifying that uses of the theme color value hyperlink must be mapped to the theme color accent1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
t1 (Text 1 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the t1 theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color t1 mapped to the theme color lt1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:t1="light1" />` The t1 attribute has a value of light1, specifying that uses of the theme color value t1 must be mapped to the theme color lt1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).
t2 (Text 2 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the t2 theme color shall be used. [*Example*: Consider a WordprocessingML document that must have references to the theme color t2 mapped to the theme color dk1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings: `<w:clrSchemeMapping w:t2="dark1" />` The t2 attribute has a value of dark1, specifying that uses of the theme color value t2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	must be mapped to the theme color dk1. *end example*] The possible values for this attribute are defined by the ST_WmlColorSchemeIndex simple type (§17.18.103).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorSchemeMapping) is located in §A.1. *end note*]

17.15.1.21 compat (Compatibility Settings)

This element specifies a set of optional compatibility options for the current document.

[*Example*: Consider a WordprocessingML document with a series of compatibility settings:

```
<w:settings>
  <w:compat>
    …
  </w:compat>
</w:settings>
```

The compat element specifies the set of compatibility settings for a document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Compat) is located in §A.1. *end note*]

17.15.1.22 consecutiveHyphenLimit (Maximum Number of Consecutively Hyphenated Lines)

This element specifies the maximum number of consecutive lines of text that can end with a hyphen when the contents of this document are displayed. Once this limit has been reached, the following line shall not be hyphenated regardless of whether or not it meets the criteria needed for hyphenation.

If this element is omitted or has its val attribute equal to `0`, the given WordprocessingML document shall have no limit on the number of consecutive lines of text that can end with a hyphen.

[*Example*: Consider a WordprocessingML document which should automatically be hyphenated. If the contents of this document result in hyphens appearing on every line in the document, as follows:

Test test test Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The Mis-sissippi The Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The Mis-sissippi The Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The Mis-sissippi

This output might be undesirable. If the document shall have a maximum of two consecutive hyphens, this requirement is specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:consecutiveHyphenLimit w:val="2" />
```

The consecutiveHyphenLimit element's val attribute has a value of 2 specifying that a maximum of two hyphens should be allowed, limiting the hyphenation output like this:

Test test test Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The Mis-
sissippi The Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The Mis-
sissippi The Mississippi The Mississippi The Mississippi The Mississippi The Mississippi The
Mississippi

end example]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.15.1.23 decimalSymbol (Radix Point for Field Code Evaluation)

This element specifies the character that shall be interpreted as the radix point when evaluating the contents of all fields in the current document.

[*Rationale*: When evaluating field instructions based on the contents of the current document, it is necessary to know the character which must be treated as the radix point in order to prevent changes to the calculation of the same field instructions based on the current user's locale. This element stores the radix point which must be used to evaluate fields in the contents of this document, irrespective of the locale of the application loading the file. *end rationale*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, the application shall use the default radix point of its current locale setting to evaluate field instructions. If this element's attribute value is more than a single character, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document which should use the comma character as the radix point for all field instructions. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:decimalSymbol w:val="," />
```

The decimalSymbol element's val attribute has a value of `,` specifying that the comma character must be interpreted as the radix point.

For instance, the string `12.345,00` would be interpreted as a numeric value of twelve thousand three hundred and forty five. If the decimalSymbol was a period, the same string would be twelve and three hundred and forty five thousandths. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.24 defaultTableStyle (Default Table Style for Newly Inserted Tables)

This element specifies the table style which shall automatically be applied to the table properties of tables added to this document by an application. Note that it does not change the table style applied to tables which do not reference a style, instead, it automatically applies the style to that table via the tblStyle element (§17.4.62). This link is made by referencing the styleId attribute value of the table style which shall be used to format newly inserted tables.

If this element is omitted, then no table style shall automatically be applied to inserted tables (therefore inheriting the default table style). If the referenced style is not present or not a table style, then no table style shall automatically be applied to inserted tables.

[*Example*: Consider a WordprocessingML document which should use the LightShading-Accent3 style. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:defaultTableStyle w:val="LightShading-Accent3" />
```

The corresponding table style must therefore exist in the styles part:

```
<w:style w:type="table" w:styleId="LightShading-Accent3">
  …
</w:style>
```

The defaultTableStyle element's val attribute has a value of `LightShading-Accent3` specifying that that style is applied automatically to newly inserted tables. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.1.25 defaultTabStop (Distance Between Automatic Tab Stops)

This element specifies the value which shall be used as the multiplier to generate automatic tab stops in this document. *Automatic tab stops* refer to the tab stop locations which occur after all custom tab stops in the current paragraph have been surpassed.

If this element is omitted, then automatic tab stops should be generated at 720 twentieths of a point (0.5") intervals across the displayed page.

[*Example*: Consider a WordprocessingML document which should have automatic tab stops every 360 twentieths of a point (0.25 inches). This requirement is specified using the following WordprocessingML in the document settings:

```
<w:defaultTabStop w:val="360" />
```

The defaultTabStop element's val attribute has a value of 360 specifying that automatic tab stops shall occur every 1/4th of an inch across the page.

If a custom tab stop was located at 2.28", then the next three automatic tab stops would be at 2.5", 2.75" and 3.0" (the next three multiples of the default tab stop value). *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />` The val attribute has a value of 720, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

17.15.1.26 displayBackgroundShape (Display Background Objects When Displaying Document)

This element specifies whether the images and colors defined in the document's background using the background element (§17.2.1) shall be displayed when the document is displayed in print layout view (§17.18.102) as specified in the view element (§17.15.1.92).

If this element is omitted, then background shapes shall not be displayed when the document is displayed in print layout view.

[*Example*: Consider a WordprocessingML document that has a turquoise background specified for all pages and is being displayed in page layout view, as follows:

If the document's background should not be displayed, that requirement would be specified using the following WordprocessingML in the document settings:

```
<w:displayBackgroundShape w:val="true" />
```

The resulting document would display the background in page layout view:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.27 displayHorizontalDrawingGridEvery (Distance between Horizontal Gridlines)

This element specifies the number of horizontal grid units defined using the drawingGridHorizontalSpacing element (§17.15.1.45) which shall be allowed between subsequent visible horizontal drawing grid lines in this document, if gridlines are being shown. [*Note*: The display of gridlines is an application-level setting not specified in ISO/IEC 29500. *end note*] The *drawing grid* is a grid which can be used by applications to help position floating objects in the document.

If this element is omitted, then gridlines shall be displayed for each horizontal grid unit.

[*Example*: Consider the image below illustrating a WordprocessingML document in which all horizontal grid units are visible (the default setting):

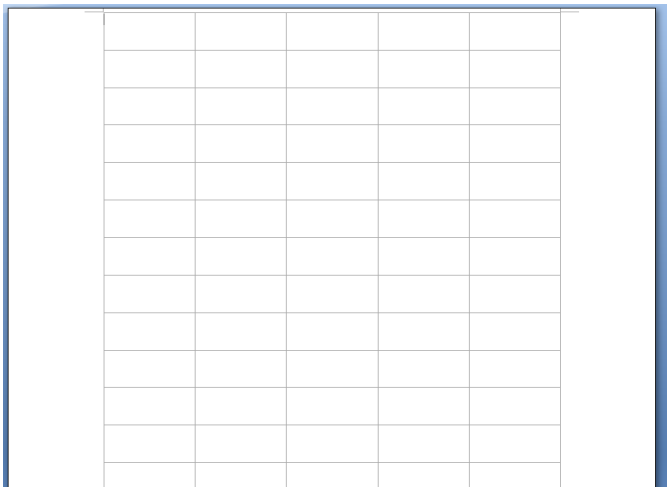
If the gridlines in this document shall only be displayed for every 4th horizontal drawing gridline, that requirement would be specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:displayHorizontalDrawingGridEvery w:val="4" />
```

The resulting grid would look like the following:



The displayHorizontalDrawingGridEvery element has its val attribute equal to 4, therefore every fourth gridline is displayed in the document when the drawing grid is turned on. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.15.1.28 displayVerticalDrawingGridEvery (Distance between Vertical Gridlines)

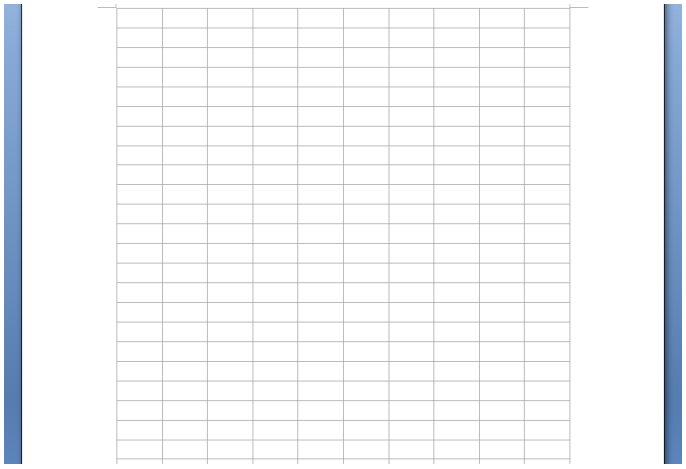
This element specifies the number of vertical grid units defined using the drawingGridVerticalSpacing element (§17.15.1.47) which shall be allowed between subsequent vertical gridlines in this document, if gridlines are being shown. [*Note*: The display of gridlines is an application-level setting not specified in ISO/IEC 29500. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

note] The *drawing grid* is a grid which can be used by applications to help position floating objects in the document.

If this element is omitted, then vertical gridlines shall not be displayed.

[*Example*: Consider the image below illustrating a WordprocessingML document in which all vertical grid units are visible (the default setting):



If the vertical drawing gridlines in this document must only be displayed for every 4th gridline, that requirement would be specified using the following WordprocessingML in the document settings:

```
<w:displayVerticalDrawingGridEvery w:val="4" />
```

The resulting grid would look like the following:

The displayVerticalDrawingGridEvery element has its val attribute equal to 4, therefore every fourth vertical gridline is displayed in the document when the drawing grid is turned on. *end example*]

Attributes	Description
val (Decimal	Specifies that the contents of this attribute contains a decimal number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Number Value)	The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.15.1.29 documentProtection (Document Editing Restrictions)

This element specifies the set of document protection restrictions which have been applied to the contents of a WordprocessingML document. These restrictions should be enforced by applications editing this document when the enforcement attribute is turned on, and ignored (but persisted) otherwise. *Document protection* is a set of restrictions used to prevent unintentional changes to all or part of a WordprocessingML document. [*Note*: This protection does not encrypt the document, and malicious applications might circumvent its use. This protection is not intended as a security feature. *end note*]

If this element is omitted, then no protection shall be applied to this document. When a password is to be hashed and stored in this element, it shall be hashed as defined below, starting from a UTF-16LE encoded string value. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
algorithmName (Cryptographic Algorithm Name)	Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value. The following values are reserved:

Value	Algorithm
MD2	Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td></td><td>[*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr></table> [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `< … algorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The algorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	example] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
edit (Document Editing Restrictions)	Specifies the set of editing restrictions which shall be enforced on a given WordprocessingML document, as defined by the simple type referenced below If this attribute is omitted, the consumer shall behave as though there are no editing restrictions applied to this document; equivalent to an attribute value of none. [*Example*: Consider a WordprocessingML document that contains the following WordprocessingML specifying that hosting applications must enforce read-only protection for a given document: `<w:documentProtection w:edit="readOnly" w:enforcement="1" />` The edit attribute has a value of `readOnly` and a enforcement attribute with a value of `1`, specifying that read-only document protection must be enforced on the given document. e*nd example*] The possible values for this attribute are defined by the ST_DocProtect simple type (§17.18.18).
enforcement (Enforce Document Protection Settings)	Specifies if the document protection settings shall be enforced for a given WordprocessingML document. If the value of this element is `off`, `0`, or `false`, all the WordprocessingML pertaining to document protection is still preserved in the document, but is not enforced. If the value of this element is on, 1, or `true`, the document protection is enforced. If this attribute is omitted, then document protection settings shall not be enforced by applications. [*Example*: Consider a WordprocessingML document that contains the following WordprocessingML specifying that hosting applications must apply read-only protection for a given document: `<w:documentProtection w:edit="readOnly" w:enforcement="1" />` The enforcement attribute has a value of 1, specifying that the document protection specified must be enforced on the given document. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
formatting (Only Allow Formatting With Unlocked Styles)	Specifies if formatting restrictions are in effect for a given WordprocessingML document. This enables the document to restrict the types of styles that can exist in a given WordprocessingML document. Specifically, by setting this attribute's value equal to `true`, every style whose locked element (§17.7.4.7) has a value of `true` (or latent styles

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	(§17.7.4.5) whose locked attribute is `true`) shall not be available for use in the application, nor should any direct formatting. Only styles with a locked value of `false` can be used.
	
	If this attribute is omitted, then no formatting restrictions shall be applied, even when document protection is enforced.
	
	[*Example*: Consider a WordprocessingML document that must apply formatting protection. This requirement would be specified using the following WordprocessingML in the document settings:
	
	`<w:documentProtection w:formatting="true" w:enforcement="true" />`
	
	If the following definition for a style was also present in the document:
	
	`<w:style w:type="paragraph" w:styleId="Heading1">`
	`<w:name w:val="Heading 1" />`
	`<w:locked w:val="1" />`
	`…`
	`</w:style>`
	
	The formatting attribute has a value of `true` specifying that the applications must not allow the style above to be added to the WordprocessingML document. This does not preclude previous uses of that style (which must not be removed), but does prevent new uses of this style from being added. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

hashValue (Password Hash Value)	Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.
	
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements:
	
	`<… AlgorithmName="SHA-1"`
	`hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hashValue attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of `SHA-1`) and that the resulting has value must be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	9oN7nWkCAyEZib1RomSJTjmPpCY= for the protection to be disabled. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: <… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The saltValue attribute value of ZUdHa+D8F/OAKP3I7ssUnQ== specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: <… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The spinCount attribute value of 100000 specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	hashValue attribute. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocProtect) is located in §A.1. *end note*]

17.15.1.30 documentType (Document Classification)

This element specifies the classification of a given WordprocessingML document.

[*Note*: This element can be used by hosting applications to facilitate customized user interface and/or automatic formatting behaviors based on the 'type' of a given WordprocessingML document. *end note*]

If this element is omitted, then the document shall be classified as a general document.

[*Example*: Consider a set of WordprocessingML documents which should be classified as 'letters'. This classification would be specified using the following WordprocessingML in the document settings of these documents:

```
<w:documentType w:val="letter" />
```

The documentType element's val attribute is equal to `letter`, specifying that the hosting application must apply the behaviors it has specified for letters to the given WordprocessingML document. *end example*]

Attributes	Description
val (Document Classification Value)	Specifies the classification of the document based on the types defined in the referenced simple type definition. [*Example*: Consider a WordprocessingML document which should be classified as an e-mail message. This classification would be specified using the following WordprocessingML in the document settings: `<w:documentType w:val="eMail" />` The val attribute is equal to `eMail`, specifying that the hosting application can apply e-mail behaviors (if any) to this document. *end example*] The possible values for this attribute are defined by the ST_DocType simple type (§17.18.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocType) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.31 docVar (Single Document Variable)

This element specifies the parameters of a single document variable. A *document variable* is a storage location for arbitrary customer data in name/value pairs that is persisted in a given WordprocessingML document. Specifically, this element specifies through its name and val attributes the name and value pair for a given document variable.

[*Note*: This mechanism is maintained for legacy compatibility only, and should be avoided in favor of the custom XML data support defined in ISO/IEC 29500. *end note*]

[*Example*: Consider the following WordprocessingML fragment specifying a document variable named `example` and containing the value `example value`:

```
<w:docVars>
  <w:docVar w:name="example" w:val="example value" />
</w:docVars>
```

The docVar element defines a single document variable, named `example` using the name attribute, and assigned the value `example value` through the val attribute. *end example*]

Attributes	Description

name (Document Variable Name)	Specifies the name of the parent document variable.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a document variable:
	
	`<w:docVars>`
	`  <w:docVar w:name="example name" w:val="example value" />`
	`</w:docVars>`
	
	The name attribute specifies that the name of the document variable is `example name`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

val (Document Variable Value)	Specifies the value of the parent document variable.
	
	[*Example*: Consider the following WordprocessingML fragment specifying a document variable:
	
	`<w:docVars>`
	`  <w:docVar w:name="example name" w:val="Tristan Davis" />`
	`</w:docVars>`
	
	The val attribute specifies that the value of the document variable is `Tristan Davis`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocVar) is located in §A.1. *end note*]

17.15.1.32 docVars (Document Variables)

This element specifies the presence of document variables in a WordprocessingML. A *document variable* is a storage location for arbitrary customer data in name/value pairs that is persisted in a given WordprocessingML document.

[*Note*: This mechanism is maintained for legacy compatibility only, and should be avoided in favor of the custom XML data support defined in ISO/IEC 29500. *end note*]

[*Example*: Consider the following WordprocessingML specifying three document variables:

```
<w:docVars>
  <w:docVar … />
  <w:docVar … />
  <w:docVar … />
</w:docVars>
```

The docVars element contains three child elements each defining a single document variable in this document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocVars) is located in §A.1. *end note*]

17.15.1.33 doNotAutoCompressPictures (Do Not Automatically Compress Images)

This element specifies that pictures in this document shall not automatically be compressed when saving the document in order to reduce the overall size of the resulting WordprocessingML document.

If this element is omitted, applications can perform basic compression on images before saving the contents of the document.

[*Example*: Consider a WordprocessingML document which should never have its images compressed before they are saved. This requirement would be specified using the following WordprocessingML:

```
<w:doNotAutoCompressPictures w:val="true"/>
```

The doNotAutoCompressPictures element's val attribute has a value of `true` specifying that images must not be automatically compressed when the document is saved. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.34 doNotDemarcateInvalidXml (Do Not Show Visual Indicator For Invalid Custom XML Markup)

This element specifies whether a visual cue should be displayed around content contained in a WordprocessingML document which is contained with custom XML markup specified via the customXml element when an application determines that the current XML markup (or its contents) violate the constraints of the attached XML schema(s).

If this element is not present in a WordprocessingML document visual cues shall be displayed on content contained in custom XML markup in a WordprocessingML document which is considered to be invalid based on the associated XML schema(s).

[*Example*: Consider a WordprocessingML document which should show no visual indication of invalid custom XML markup. This requirement would be specified using the following WordprocessingML:

```
<w:doNotDemarcateInvalidXml w:val="true"/>
```

The doNotDemarcateInvalidXml element's val attribute has a value of true specifying the display of any visual indication of invalid custom XML markup must be suppressed for this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.35 doNotDisplayPageBoundaries (Do Not Display Visual Boundary For Header/Footer or Between Pages)

This element specifies whether applications displaying this document should display the contents of the header and footer when displaying the document in print layout view (§17.18.102) or should collapse those areas as well as the whitespace on all displayed pages so that the text extents are directly following one another. [*Rationale*: Collapsing the ends of pages makes it easier to read the contents of the document, since the text flows between pages without whitespace, while maintaining the WYSIWYG functionality of print layout view for the document's main content. *end rationale*]

If this element is omitted, then all pages should be shown at their full size (including whitespace and headers/footers) when they are displayed in print layout view.

[*Example*: Consider the images below illustrating two pages in a WordprocessingML document:

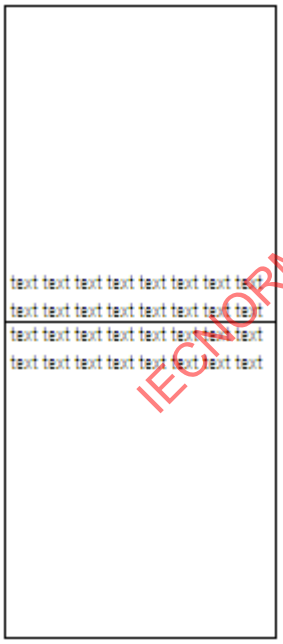
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If document must automatically have whitespace between pages removed when it is displayed, that requirement would be specified using the following WordprocessingML in the document settings:

```
<w:doNotDisplayPageBoundaries w:val="true" />
```

The resulting output might look like the following:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The doNotDisplayPageBoundaries element has its val attribute equal to `true`, therefore the document is automatically displayed with whitespace between text extents on following pages compressed, allowing the pages to be viewed more easily. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.36 doNotEmbedSmartTags (Remove Smart Tags When Saving)

This element specifies if any smart tags specified using the smartTag element shall be removed from the contents of this document before it is resaved. This setting shall also prevent the addition of new smart tags to the content of the document.

If this element is omitted, then smart tags shall not be removed from the file when it is saved.

[*Example*: Consider a WordprocessingML document which should never be saved with smartTag elements in its contents. This requirement is specified using the following WordprocessingML fragment in the document settings:

```
<w:doNotEmbedSmartTags w:val="true"/>
```

The doNotEmbedSmartTags element's val attribute has a value of true specifying that smart tags must never be saved in the contents of this document. For example, if a run formerly looked like this:

```
<w:p>
  <w:r>
    <w:t xml:space="preserve">Hello</w:t>
  </w:r>
  <w:smartTag … >
    <w:r>
      <w:t>world</w:t>
    </w:r>
  </w:smartTag>
</w:p>
```

The presence of this element specifies that the SmartTag element must be removed, and applications might then choose to combine duplicated runs as desired. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.37 doNotHyphenateCaps (Do Not Hyphenate Words in ALL CAPITAL LETTERS)

This element specifies whether or not words comprised of all capital letters shall be hyphenated within a given document when automatic hyphenation is specified via the autoHyphenation element (§17.15.1.10).

If this element is omitted, then words in ALL CAPITAL LETTERS shall be hyphenated when the document is hyphenated.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a document which is automatically hyphenated containing the following paragraph of content:

THIS IS A HYPHENATION EXAMPLE. THIS IS A SHORT HYPHENATION EXAMPLE. THIS IS A SHORT HYPHE-NATION EXAMPLE.

If words in ALL CAPITAL LETTERS must not be hyphenated, this requirement would be specified by adding the following WordprocessingML to the document settings part:

```
<w:doNotHyphenateCaps w:val="true"/>
```

The resulting content would not be hyphenated:

THIS IS A HYPHENATION EXAMPLE. THIS IS A SHORT HYPHENATION EXAMPLE. THIS IS A SHORT HYPHENATION EXAMPLE.

The doNotHyphenateCaps element val set to `true`, specifying that the first line of text to end with the word SHORT as the word HYPHENATION had to be moved to the second line since it could not fit in its entirety on the first line.

Conversely, setting the doNotHyphenateCaps element val set to `off` (the default) caused the first line of text to contain a hyphenated portion of the word HYPHENATION as hyphenation of words comprised of all capital letters is permitted. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.38 doNotIncludeSubdocsInStats (Do Not Include Content in Text Boxes, Footnotes, and Endnotes in Document Statistics)

This element specifies if document content contained in text boxes, footnotes, and endnotes shall be excluded when an application calculates a given document's statistics when these values are calculated and/or displayed by an application.

[*Note*: Some examples of document statistics that an application might chose to calculate are: number of words, number of characters, number of paragraphs, number of pages, number of lines, and so on. *end note*]

[*Example*: Consider a WordprocessingML that specifies that it must not include these document stories when its contents are used to calculate document statistics. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:doNotIncludeSubdocsInStats w:val="true"/>
```

The doNotIncludeSubdocsInStats element's val attribute has a value of `true` specifying that only the contents of the main document story should be used when calculating document statistics. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.39 doNotShadeFormData (Do Not Show Visual Indicator For Form Fields)

This element specifies whether a visual cue should be displayed around form fields contained in a WordprocessingML document specified via the `FORMTEXT`, `FORMCHECKBOX`, or `FORMDROPDOWN` fields.

If this element is not present in a WordprocessingML document visual cues should be displayed on form fields contained in the document.

[*Example*: Consider a WordprocessingML document which should no visual indication of form fields. This requirement would be specified using the following WordprocessingML:

```
<w:doNotShadeFormData w:val="true"/>
```

The doNotShadeFormData element's val attribute has a value of `true` specifying the display of any visual indication of form fields must be suppressed for this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.40 doNotTrackFormatting (Do Not Track Formatting Revisions When Tracking Revisions)

This element specifies that applications shall not track revisions made to the formatting of this WordprocessingML document when the trackRevisions element (§17.15.1.89) is turned on.

If this element is omitted, then revisions to formatting shall be generated by changes to the contents of this document when the trackRevisions element is turned on.

[*Example*: Consider a WordprocessingML document containing the text run `Example` that must have revisions tracked. Example WordprocessingML from Document 1 is given below:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

If the word `text` was added to the end of this document and bolded, the resulting WordprocessingML would be output as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
      <w:ins … >
        <w:r>
          <w:rPr>
            <w:b/>
            <w:rPrChange … >
              <w:rPr/>
            </w:rPrChange>
          </w:rPr>
          <w:t>text</w:t>
        </w:r>
      </w:ins>
    </w:p>
  </w:body>
</w:document>
```

If changes to formatting were turned off using the following WordprocessingML syntax in the document settings:

```
<w:settings>
  <w:trackRevisions w:val="true" />
  <w:doNotTrackFormatting w:val="true" />
  …
</w:settings>
```

The same revision (the word text was added to the end of this document and bolded) would result in the following markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
      <w:ins … >
        <w:r>
          <w:rPr>
            <w:b/>
          </w:rPr>
          <w:t>text</w:t>
        </w:r>
      </w:ins>
    </w:p>
  </w:body>
</w:document>
```

The doNotTrackFormatting element's val attribute was set to `true`, therefore the changes to the formatting of the document were not tracked as revisions in the document's WordprocessingML. Specifically, applying bold formatting to the text was not tracked as a revision with the rPrChange (§17.13.5.31) element. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.41 doNotTrackMoves (Do Not Use Move Syntax When Tracking Revisions)

This element specifies that applications shall not track revisions made to this WordprocessingML document as moves when the trackRevisions element (§17.15.1.89) is turned on, even when that syntax is appropriate. Instead, applications should use a standard insertion and deletion annotation syntax. Existing moves shall not be modified. [*Rationale*: This element is provided to enable interoperability with earlier word processing applications which do not understand moves. *end rationale*]

If this element is omitted, then move annotations can be generated by changes to the contents of this document when the trackRevisions element is turned on as appropriate.

[*Example*: Consider a WordprocessingML that specifies that it must not have additional moves added to its contents. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:doNotTrackMoves w:val="true"/>
```

The doNotTrackMoves element's val attribute has a value of `true` specifying that insertion/deletion annotations must be used rather than moves when revisions are tracked in this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.42 doNotUseMarginsForDrawingGridOrigin (Do Not Use Margins for Drawing Grid Origin)

This element specifies that the top-left corner of the page shall not be used as the origin for the drawing grid. The *drawing grid* is a virtual grid which can be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects are aligned, etc.). If this element is present the grid shall start at the top-left edge of the page and not the text extents.

If this element is omitted, then the gridlines shall start at the topmost edge of the text extents.

[*Example*: Consider a WordprocessingML document whose drawing grid must begin at the top left edge of the page. This requirement would be specified using the following WordprocessingML markup in the document settings:

```
<w:doNotUseMarginsForDrawingGridOrigin w:val="true" />
```

The doNotUseMarginsForDrawingGridOrigin element's val attribute is equal to `true` specifying that the document's drawing grid must begin from the top left corner of the page, rather than the top left corner of the text extents. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.43 doNotValidateAgainstSchema (Do Not Validate Custom XML Markup Against Schemas)

This element specifies that applications shall not validate the custom XML markup in this document against the applicable custom XML schema(s), even when those schemas are available. The application should silently behave as if it was unable to provide this functionality.

If this element is omitted, then applications which support this functionality should attempt to validate the custom XML contents against any available related custom XML schema(s).

[*Example*: Consider a WordprocessingML document which should not have its custom XML content validated even by applications which support this operation. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:doNotValidateAgainstSchema w:val="true" />
```

The doNotValidateAgainstSchema element's val attribute has a value of `true` specifying that the custom XML markup in this document must not be validated. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.44 drawingGridHorizontalOrigin (Drawing Grid Horizontal Origin Point)

This element specifies the distance from of the left edge of the page which shall be used as the origin for the horizontal gridlines used by the drawing grid. The *drawing grid* is a virtual grid which might be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

are aligned, etc.). Since the grid always covers the entire page when the doNotUseMarginsForDrawingGridOrigin element (§17.15.1.42) is specified, this element shall only affect the starting edge of the first horizontal gridline displayed (i.e. it only adjusts the grid by the modulus of the value against the width of one grid unit).

If this element is omitted, then the gridlines shall start at the leftmost edge of the page. If the doNotUseMarginsForDrawingGridOrigin element is not specified, then this element is ignored.

[*Example*: Consider a WordprocessingML document whose drawing grid must begin three inches (4320 twentieths of a point) before the left edge of the page. This requirement would be specified using the following WordprocessingML markup in the document settings:

```
<w:settings>
  …
  <w:doNotUseMarginsForDrawingGridOrigin w:val="true" />
  <w:drawingGridHorizontalOrigin w:val="4320" />
  …
</w:settings>
```

The drawingGridHorizontalOrigin element's val attribute is equal to `4320` specifying that the horizontal edge of the document's drawing grid must begin three inches (4320 twentieths of a point) from the left edge of the page, since the doNotUseMarginsForDrawingGridOrigin element's val attribute is equal to `true`. *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />` The val attribute has a value of `720`, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

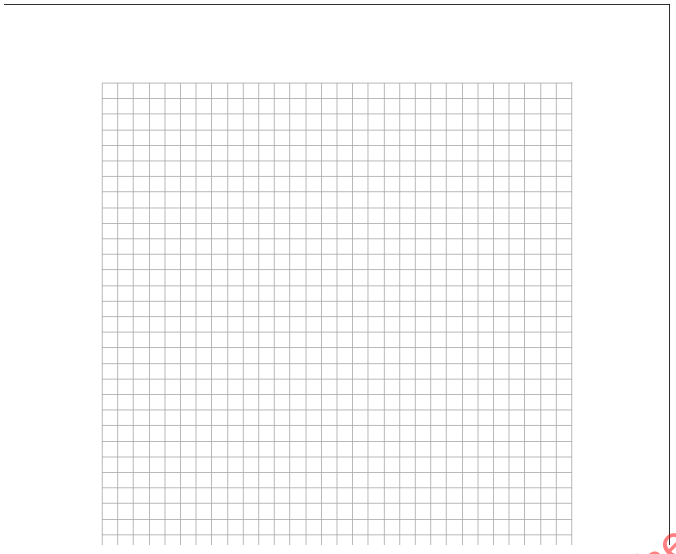
©ISO/IEC 2016 – All rights reserved

17.15.1.45 drawingGridHorizontalSpacing (Drawing Grid Horizontal Grid Unit Size)

This element specifies the width of horizontal grid units in this document. The *drawing grid* is a grid which can be used by applications to help position floating objects in the document.

If this element is omitted, then each horizontal grid unit shall be 180 twentieths of a point (0.125") in width.

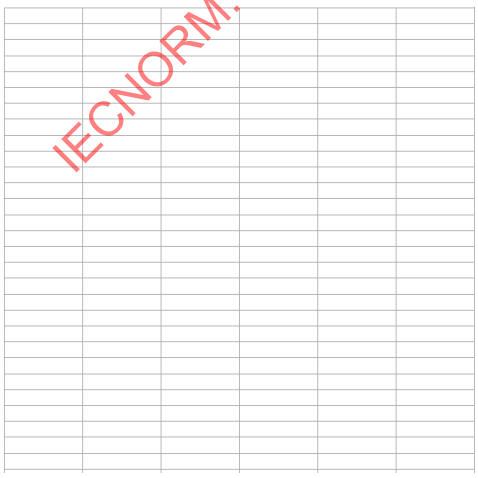
[*Example*: Consider the image below illustrating a WordprocessingML document in which all horizontal grid units are each 144 twentieths of a point wide (and all are showing):



If the gridlines in this document must only be displayed for every half an inch, that requirement would be specified using the following WordprocessingML in the document settings:

```
<w:drawingGridHorizontalSpacing w:val="720" />
```

The resulting grid would look like the following:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The drawingGridHorizontalSpacing element has its val attribute equal to 720, therefore every horizontal gridline has a width of one half of an inch (720 twentieths of a point). *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />` The val attribute has a value of 720, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

17.15.1.46 drawingGridVerticalOrigin (Drawing Grid Vertical Origin Point)

This element specifies the distance from of the top edge of the page which shall be used as the origin for the vertical gridlines used by the drawing grid. The *drawing grid* is a virtual grid which can be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects are aligned, etc.). Since the grid always covers the entire page when the doNotUseMarginsForDrawingGridOrigin element (§17.15.1.42) is specified, this element shall only affect the starting edge of the first vertical gridline displayed (i.e. it only adjusts the grid by the modulus of the value against the width of one grid unit).

If this element is omitted, then the gridlines shall start at the topmost edge of the page. If the doNotUseMarginsForDrawingGridOrigin element is not specified, then this element is ignored.

[*Example*: Consider a WordprocessingML document whose drawing grid must begin one inch (1440 twentieths of a point) before the top edge of the page. This requirement would be specified using the following WordprocessingML markup in the document settings:

```
<w:settings>
  …
  <w:doNotUseMarginsForDrawingGridOrigin w:val="true" />
  <w:drawingGridVerticallOrigin w:val="1440" />
  …
</w:settings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The drawingGridVerticalOrigin element's val attribute is equal to 1440 specifying that the vertical edge of the document's drawing grid must begin one inch (1440 twentieths of a point) from the top edge of the page, since the doNotUseMarginsForDrawingGridOrigin element's val attribute is equal to true. *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />` The val attribute has a value of 720, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

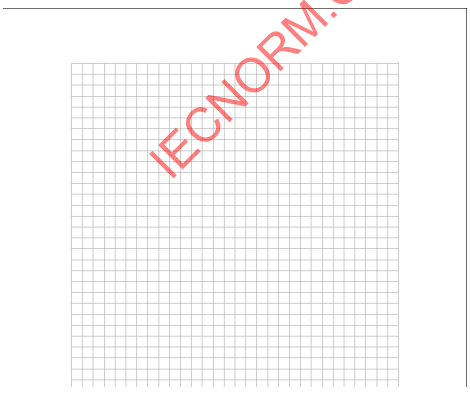
[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

17.15.1.47 drawingGridVerticalSpacing (Drawing Grid Vertical Grid Unit Size)

This element specifies the width of vertical grid units in this document. The *drawing grid* is a grid which can be used by applications to help position floating objects in the document.

If this element is omitted, then each vertical grid unit shall be 180 twentieths of a point (0.125") in width.

[*Example*: Consider the image below illustrating a WordprocessingML document in which all vertical grid units are each 144 twentieths of a point high (and all are showing):



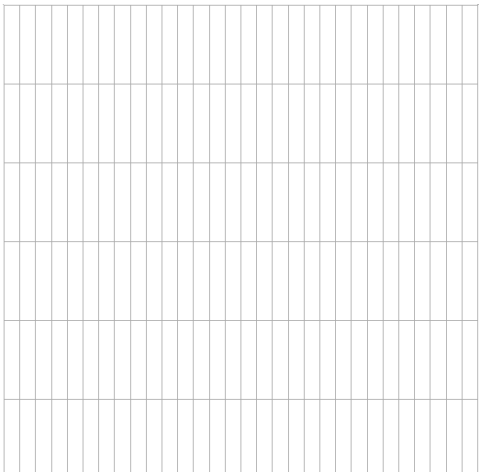
If the vertical gridlines in this document must only be displayed for every half an inch, that requirement would be specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:drawingGridVerticalSpacing w:val="720" />
```

The resulting grid would look like the following:



The drawingGridVerticalSpacing element has its val attribute equal to 720, therefore every vertical gridline has a height of one half of an inch (720 twentieths of a point). *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />` The val attribute has a value of 720, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.48 forceUpgrade (Upgrade Document on Open)

This element specifies that the contents of this document can be upgraded and that the resulting document shall not have its functionality limited to only those functions compatible with earlier word processing applications. The only actions required as part of upgrading the document are:

- The removal of this element. If an application does not know how to upgrade a document, this element should be ignored and persisted.
- The removal of all compatibility options (§17.15.1.21) on the document which maintain compatibility with previous word processing applications. The compatibility settings which simply affect a given behavior shall not be turned off.

[*Note*: The remaining operations which must be performed as part of upgrading the document are application-defined and outside the scope of ISO/IEC 29500. *end note*]

[*Example*: Consider a WordprocessingML document that specifies that it must automatically be upgraded when it is opened by an application. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:forceUpgrade/>
```

The forceUpgrade element's presence specifies that this document must be upgraded by any application which supports this operation. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.1. *end note*]

17.15.1.49 formsDesign (Structured Document Tag Placeholder Text Should be Resaved)

This element specifies that the document was last saved while the placeholder text of all structured document tags in this document were being edited. This means that the placeholder text currently displayed in all structured document tags which are displaying the showingPlcHdr element (§17.5.2.39) shall be committed to the corresponding glossary document entry as specified using the docPart element (§17.12.5) when this document is opened, in order to ensure that the most recent placeholder text is stored in the glossary document entry. If the current placeholder text cannot be saved as a glossary document entry, then it should be modified as needed before saving.

If this element is omitted, then the placeholder text in this document should not automatically be resaved when the document is opened.

[*Example*: Consider a WordprocessingML document that specifies that its placeholder text should be resaved to the glossary document when the file is opened. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:formsDesign w:val="true"/>
```

The formsDesign element's val attribute has a value of `true` specifying that this document should be resaved to its glossary document by any application which supports this operation. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.50 gutterAtTop (Position Gutter At Top of Page)

This element specifies that a given WordprocessingML document's gutter shall be positioned at the top of the document's pages when the document is displayed. A *gutter* is the white space formed by the inner margins of two pages facing one another; such as the white space between the text on pages of a book when the book is opened.

If this element is omitted, then the gutter shall not be positioned at the top of the page. If the mirrorMargins (§17.15.1.57), bookFoldPrinting (§17.15.1.11), bookFoldRevPrinting (§17.15.1.13), or printTwoOnOne (§17.15.1.64) elements are used within a given document, the gutterAtTop element shall not be used. Rather, the gutter shall be positioned automatically as necessary to enable the printing and page layout capabilities of these settings.

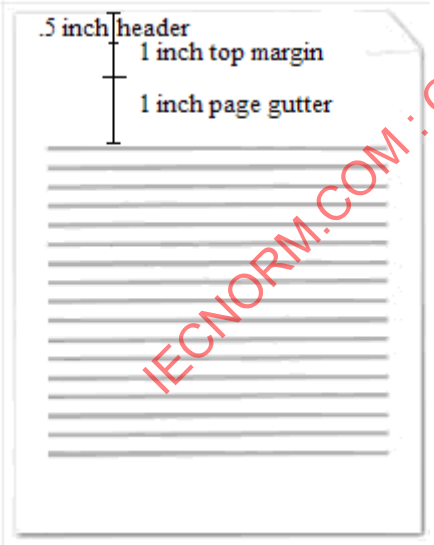
[*Example*: Consider a one page WordprocessingML document with a 1,440 twentieths of a point (one inch) top margin and gutter, and a 720 twentieths of a point (one half of an inch) header. Consider also, that the gutter must exist at the top of the document's pages. This requirement is specified using the following WordprocessingML in the section properties:

```
<w:pgMar w:top="1440" … w:header="720" … w:gutter="1440" />
```

And the following WordprocessingML in the document settings:

```
<w:gutterAtTop w:val="true" />
```

The resulting document's pages would have the gutter positioned as follows:



The gutterAtTop element's val attribute is equal to `true`, specifying that the gutter must appear at the top of each page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.1.51 hideGrammaticalErrors (Do Not Display Visual Indication of Grammatical Errors)

This element specifies whether a visual cue should be displayed around run content contained in a WordprocessingML document which has been flagged as a possible grammatical error using the proofErr element (§17.13.8.1) or via the application's own grammar engine.

If this element is not present in a WordprocessingML document, visual cues shall be displayed on content contained in a WordprocessingML document which is considered to contain grammatical errors.

[*Example*: Consider a WordprocessingML document which should show no visual indication of grammatical errors. This requirement would be specified using the following WordprocessingML:

```
<w:hideGrammaticalErrors w:val="true"/>
```

The hideGrammaticalErrors element's val attribute has a value of `true` specifying the display of any visual indication of grammatical errors must be suppressed for this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.52 hideSpellingErrors (Do Not Display Visual Indication of Spelling Errors)

This element specifies whether a visual cue should be displayed around run content contained in a WordprocessingML document which has been flagged as a possible spelling error using the proofErr element (§17.13.8.1) or via the application's own spelling engine.

If this element is not present in a WordprocessingML document, visual cues shall be displayed on content contained in a WordprocessingML document which is considered to contain spelling errors.

[*Example*: Consider a WordprocessingML document which should show no visual indication of spelling errors. This requirement would be specified using the following WordprocessingML:

```
<w:hideSpellingErrors w:val="true"/>
```

The hideSpellingErrors element's val attribute has a value of `true` specifying the display of any visual indication of spelling errors must be suppressed for this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.53 hyphenationZone (Hyphenation Zone)

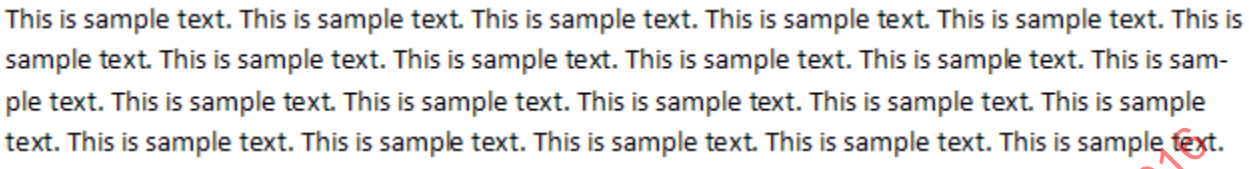
This element specifies the hyphenation zone which shall be used when automatically or manually hyphenating the contents of this document. The *hyphenation zone* is the amount of whitespace which can be left at the end of a line (or added to justified lines) before hyphenation should be attempted on the next word in the document (in order to reduce the amount of whitespace on the line). A smaller hyphenation zone should reduce the raggedness of the right edge of a given document's body text, as more words is hyphenated. Conversely, a larger hyphenation zone should increase the raggedness of the right edge of a given document's text, as fewer words is hyphenated.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then a default hyphenation zone of 360 twentieths of a point (0.25") shall be applied when performing hyphenation on this document.

[*Example*: Consider the images below illustrating a paragraph of text in a WordprocessingML document which was automatically hyphenated with the default hyphenation zone:


This is sample text. This is sample text. This is sample text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

17.15.1.54 ignoreMixedContent (Ignore Mixed Content When Validating Custom XML Markup)

This element specifies that applications should ignore all text content which is not contained within a leaf custom XML markup element when validating the contents of the custom XML markup in this document against one or more attached custom XML schema(s). A *leaf element* is a custom XML element which has no child custom XML elements (it is a leaf in the custom XML tree).

If this element is omitted, then text content in leaf elements shall not be ignored when validating the custom XML markup against one or more custom XML schema(s).

[*Example*: Consider a WordprocessingML document that contains the following markup:

```
<w:customXml w:element="invoice" w:uri="http://www.example.com/invoice">
  <w:p>
    <w:r>
      <w:t>Invoice #:</w:t>
    </w:r>
    <w:customXml w:element="id" w:uri="http://www.example.com/invoice">
      <w:r>
        <w:t>012345</w:t>
      </w:r>
    </w:customXml>
  </w:p>
  <w:p>
    <w:r>
      <w:t>Invoice Date:</w:t>
    </w:r>
    <w:customXml w:element="date" w:uri="http://www.example.com/invoice">
      <w:r>
        <w:t>01/29/2009</w:t>
      </w:r>
    </w:customXml>
  </w:p>
</w:customXml>
```

If all the custom markup is extracted from the document, that markup would include all content in the document, i.e.:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<invoice xmlns="http://www.example.com/invoice">
  Invoice #:
  <id>012345</id>
  Invoice Date
  <date>01/29/2009</date>
</invoice>
```

The content shown above is formatted for readability. In fact, there is no such space in the XML.

However, if the ignoreMixedContent element is present with a val attribute value of `true` (or equivalent) then an application should ignore all text nodes in elements with mixed content, i.e.:

```
<invoice xmlns="http://www.example.com/invoice">
  <id>012345</id>
  <date>01/29/2009</date>
</invoice>
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.55 linkStyles (Automatically Update Styles From Document Template)

This element specifies that styles in the given document shall be updated to match the styles in the attached template specified using the attachedTemplate element (§17.15.1.6) when the document is opened by a hosting application. This setting enables the styles contained in documents with attached templates to stay synchronized with the styles used in the attached template.

If this element is omitted, then styles shall not be updated based on the document template regardless of its availability. If the attached template cannot be located or is not a valid file, then this setting should be silently ignored.

[*Example*: Consider a WordprocessingML document which should always update its styles with those defined in the document's attached template. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:settings>
  <w:linkStyles w:val="true" />
  <w:attachedTemplate r:id="rId10" />
  …
</w:settings>
```

The linkStyles element has a val attribute value of `true`, specifying that applications should attempt to locate the document template referenced by the relationship specified in the attachedTemplate element and update the document's styles with the styles from that template. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.56 listSeparator (List Separator for Field Code Evaluation)

This element specifies the character that shall be interpreted as a list item separator when evaluating the contents of all fields in the current document.

[*Rationale*: When evaluating field instructions based on the contents of the current document, it is necessary to know the character which must be treated as the list separator in order to prevent changes to the calculation of the same field instructions based on the current user's locale. This element stores the list separator which must be used to evaluate fields in the contents of this document, irrespective of the locale of the application loading the file. *end rationale*]

If this element is omitted, the application shall use the default list separator of its current locale setting to evaluate field instructions. If this element's attribute value is more than a single character, then the document is non-conformant.

[*Example*: Consider a WordprocessingML document which should use the semicolon character as the list separator for all field instructions. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:listSeparator w:val=";" />
```

The listSeparator element's val attribute has a value of `;` specifying that the semicolon character shall be interpreted as a list item separator.

For instance, the string `10;20,5` would be interpreted as having two values - `10` and `20,5`. If the listSeparator was a comma, the same string would be interpreted as `10;20` and `5`. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.1.57 mirrorMargins (Mirror Page Margins)

This element specifies that the left and right margins defined in the section properties shall be swapped on every second page. [*Note:* Page numbering can be set arbitrarily, so the flip might not always be on the pages with even-numbered labels. *end note*]

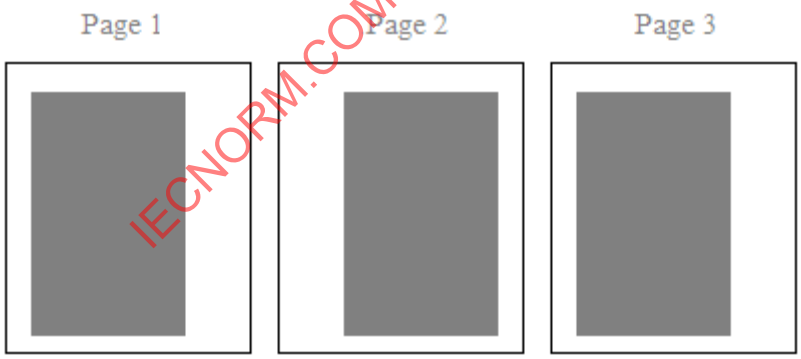
[*Guidance*: This setting is generally used when printing on both sides of pages and binding them like a book. *end guidance*]

[*Example*: Consider a graphical representation (below) of a three page WordprocessingML document with a left margin of 1" and a right margin of 2".

If the mirrorMargins element is present in the document settings with its val attribute equal to `true`, as follows:

```
<w:mirrorMargins w:val="true" />
```

The resulting pages has mirrored margins as follows (un this representation, the gray rectangles representing the text extents on each page):



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.1.58 noLineBreaksAfter (Custom Set of Characters Which Cannot End a Line)

This element specifies the set of characters which shall be restricted from ending a line for runs of text which shall be subject to custom line breaking logic using the kinsoku element (§17.3.1.16) when the contents of the document are displayed. This constraint shall only apply to text which has been flagged in the language of this rule via the lang element (§17.3.2.20) or automatic detection methods outside the scope of ISO/IEC 29500.

If this element is omitted, then no custom set of characters shall be used to restrict the characters which can end a line when using the kinsoku element.

[*Example*: Consider a paragraph of WordprocessingML text displayed as follows, with the dollar symbol $ was flagged as Japanese content using the following WordprocessingML in the run properties:

```
<w:r>
  <w:rPr>
    <w:lang w:eastAsia="ja-JP" />
  </w:rPr>
  <w:t>$</w:t>
</w:r>
```

This is a sample line of text. This is a sample line of text. This is a sample line of text. This is a $ sample line of text.

This text is displayed and the resulting first line ends with the dollar sign symbol. If this character must not be used to end a line, that requirement would be specified as follows in the document settings:

```
<w:noLineBreaksAfter w:lang="ja-JP" w:val="$" />
```

The noLineBreaksAfter element's val attribute has a value of `ja-JP`, specifying that all dollar signs in this document which are marked as Japanese text must not be allowed to end a line. This means that the dollar sign character must therefore be moved to the next line as it can no longer be the last character on a line:

This is a sample line of text. This is a sample line of text. This is a sample line of text. This is a $ sample line of text.

end example]

Attributes	Description
lang (Language For Which Custom Line Breaking Rule Applies)	Specifies the language of text for which the parent custom line breaking rule shall be applied. Applications supporting this functionality shall support custom line breaking for the following four languages: • Chinese (Traditional) • Chinese (Simplified) • Japanese • Korean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Applications can also support custom line breaking rules for other languages, but this is not required. [*Example*: Consider a WordprocessingML document which must have a custom line breaking rule for Japanese. That requirement would be specified as follows in the document settings: `<… w:lang="ja-JP" w:val="$" />` The lang attribute has a value of `ja-JP`, specifying that the rules must be applied to Japanese text. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
val (Characters For Custom Line Breaking Rule)	Specifies the set of characters which shall be included in the custom line breaking rule. [*Example*: Consider a WordprocessingML document which must have a custom line breaking rule for Japanese. That requirement would be specified as follows in the document settings: `<… w:lang="ja-JP" w:val="$" />` The val attribute has a value of `$`, specifying that the dollar sign character is the only restricted character for Japanese text. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Kinsoku) is located in §A.1. *end note*]

17.15.1.59 noLineBreaksBefore (Custom Set Of Characters Which Cannot Begin A Line)

This element specifies the set of characters which shall be restricted from beginning a new line for runs of text which shall be subject to custom line breaking logic using the kinsoku element (§17.3.1.16) when the contents of the document are displayed. This constraint shall only apply to text which has been flagged in the language of this rule via the lang element (§17.3.2.20) or automatic detection methods outside the scope of ISO/IEC 29500.

If this element is omitted, then no custom set of characters shall be used to restrict the characters which can end a line when using the kinsoku element.

[*Example*: Consider a paragraph of WordprocessingML text displayed as follows, with the dollar symbol $ was flagged as Korean content using the following WordprocessingML in the run properties:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:rPr>
    <w:lang w:eastAsia="ko-KR" />
  </w:rPr>
  <w:t>$</w:t>
</w:r>
```

This is a sample line of text. This is a sample line of text. This is a sample line of text. This is a
$ sample line of text.

This text is displayed and the resulting second line begins with the dollar sign symbol. If this character must not be used to begin a line, that requirement would be specified as follows in the document settings:

```
<w:noLineBreaksBefore w:lang="ko-KR" w:val="$" />
```

The noLineBreaksBefore element's val attribute has a value of ko-KR, specifying that all dollar signs in this document which are marked as Korean text must not be allowed to begin a line. This means that the previous word character must therefore be moved to the next line as the dollar sign can no longer be the first character on a line:

This is a sample line of text. This is a sample line of text. This is a sample line of text. This is
a $ sample line of text.

end example]

Attributes	Description
lang (Language For Which Custom Line Breaking Rule Applies)	Specifies the language of text for which the parent custom line breaking rule shall be applied. Applications supporting this functionality shall support custom line breaking for the following four languages: • Chinese (Traditional) • Chinese (Simplified) • Japanese • Korean Applications can also support custom line breaking rules for other languages, but this is not required. [*Example*: Consider a WordprocessingML document which must have a custom line breaking rule for Japanese. That requirement would be specified as follows in the document settings: `<… w:lang="ja-JP" w:val="$" />` The lang attribute has a value of ja-JP, specifying that the rules must be applied to Japanese text. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
val (Characters For Custom Line Breaking Rule)	Specifies the set of characters which shall be included in the custom line breaking rule. [*Example*: Consider a WordprocessingML document which must have a custom line breaking rule for Japanese. That requirement would be specified as follows in the document settings: `<… w:lang="ja-JP" w:val="$" />` The val attribute has a value of $, specifying that the dollar sign character is the only restricted character for Japanese text. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Kinsoku) is located in §A.1. *end note*]

17.15.1.60 noPunctuationKerning (Never Kern Punctuation Characters)

This element specifies that punctuation characters shall not be kerned in the current document when kerning is enabled on a run using the kern element (§17.3.2.19). *Kerning* refers to a process by which a hosting application shall reduce the spacing of adjacent characters and/or punctuation to improve the visual appearance of text. Well kerned text has a similar amount of blank space between each pair of characters and/or each set of a character and punctuation symbol. When kerning is enabled, Latin text shall always be kerned, and this option shall control whether punctuation characters are also kerned.

If this element is omitted, then punctuation characters shall be kerned when kerning is enabled on a given run.

[*Example*: Consider a WordprocessingML document that must not kern punctuation even when kerning is enabled on a given run. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:noPunctuationKerning w:val="true" />
```

The noPunctuationKerning element's val attribute has a value of `true`, specifying that punctuation characters must not be kerned in this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.61 printFormsData (Only Print Form Field Content)

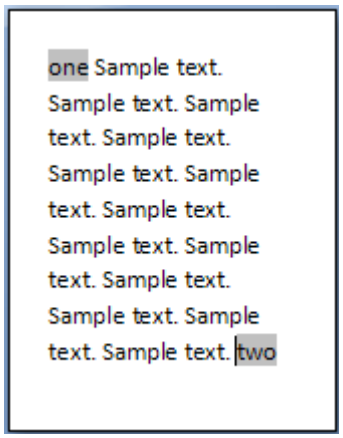
This element specifies that printing the contents of this document shall only print the contents of WordprocessingML form fields defined using the `FORMTEXT`, `FORMCHECKBOX`, and `FORMDROPDOWN` field codes in their current locations on the page - all other document contents shall be suppressed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Rationale*: This setting is typically used to allow duplication of paper forms in electronic WordprocessingML document form, allowing the resulting online document to be printed into the correct locations on the existing paper form. *end rationale*]

If this element is omitted, then the contents of the entire document (not just form fields) should be printed according to the normal print settings.

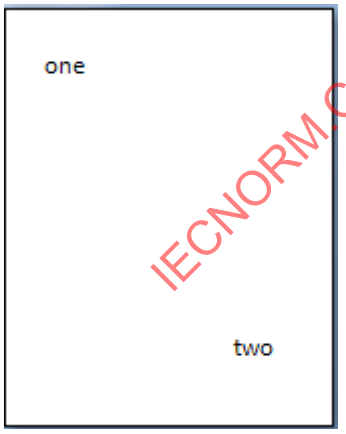
[*Example*: Consider a WordprocessingML document which has form fields in the top right and bottom left corners of the first page, as follows (with the text box form fields shaded in grey):



If the only content which must be printed on the page are the form fields' contents, this requirement is specified using the following WordprocessingML in the document settings:

```
<w:printFormsData w:val="true" />
```

The printFormsData element's val attribute as a value of `true`, specifying that only form field data shall be printed, resulting in output as follows when printed:



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.15.1.62 printFractionalCharacterWidth (Print Fractional Character Widths)

This element specifies the contents of this document shall be printed with fractional character widths. *Fractional character widths* exist when the spacing between characters is not constant (i.e. a proportional font face is used).

[*Note*: Fractional character widths are generally used in conjunction with large font sizes to prevent characters from running together or having too much space between one another. *end note*]

[*Example*: Consider a WordprocessingML document which should be printed using fractional character widths as needed. This requirement is specified using the following WordprocessingML markup in the document settings:

```
<w:printFractionalCharacterWidth w:val="true"/>
```

The printFractionalCharacterWidth element's val attribute is equal to `true`, specifying that fractional character widths can be used as necessary. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.63 printPostScriptOverText (Print PostScript Codes With Document Text)

This element specifies that the PostScript codes specified in WordprocessingML documents containing `PRINT` fields shall be included in foreground (on the same Z-order as text) with the data printed in the contents of a given WordprocessingML document.

[*Note*: This setting is maintained to ensure compatibility of legacy word processing documents. The `PRINT` field should not be used in lieu of newer technologies in ISO/IEC 29500. *end note*]

If this element is omitted, then the contents of `PRINT` fields shall be printed behind text (i.e. in the background).

[*Example*: Consider a WordprocessingML document containing `PRINT` fields whose PostScript code must be printed in the foreground of the WordprocessingML document. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:printPostScriptOverText w:val="true"/>
```

The printPostScriptOverText element's val attribute is equal to `true` specifying that the PostScript codes must be treated as results for the main text level of the document (i.e. not behind that text). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.64 printTwoOnOne (Print Two Pages Per Sheet)

This element specifies whether two pages should be printed on one sheet of paper when this document is printed. Specifically, this element specifies that each page displayed for the contents in a given WordprocessingML document should be the page size specified in the section settings divided in half with two top margins originating from the bisector of the page, and bottom margins instantiated at the top and bottom of each page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

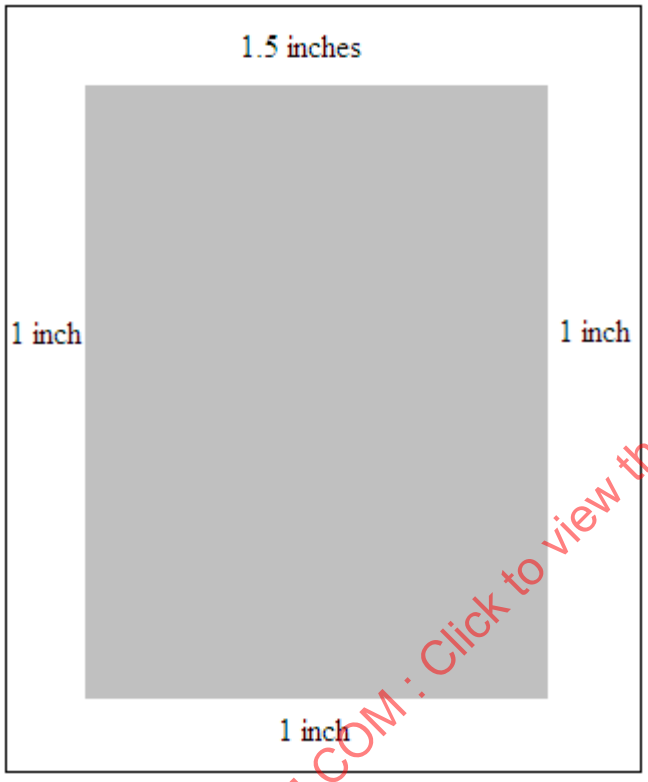
©ISO/IEC 2016 – All rights reserved

If this element is omitted, then pages should be displayed and printed as one per sheet.

[*Example*: Consider a one section document with a 2,160 twentieths of a point (one and a half inch) top margin, and 1,440 twentieths of a point (one inch) bottom, right, and left margins surrounding the document editing canvas (represented by the gray shaded area in diagrams below). This page setup is represented in WordprocessingML using the following fragment:

```
<w:pgMar w:header="0" w:top="2160" w:right="1440" w:bottom="1440" w:left="1440"
  w:footer="720" w:gutter="0" />
```

The resulting printed pages would appear as follows:



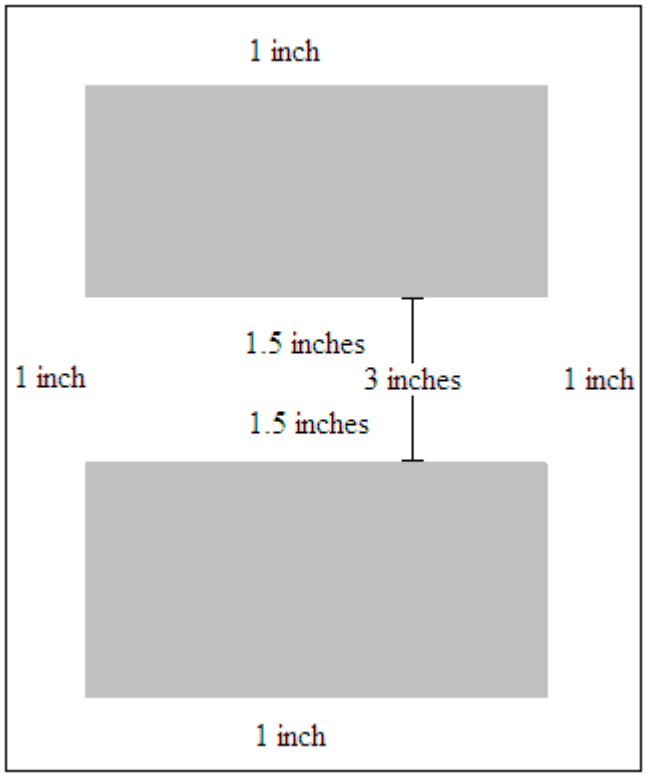
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If a document should be displayed and printed as though two pages were printed on a single sheeting, this requirement would be specified using the following WordprocessingML:

```
<w:printTwoOnOne w:val="true" />
```

The printTwoOnOne element's val attribute is equal to `true` specifying that pages should be printed two to a sheet, resulting in the following printout given these page margins:

©ISO/IEC 2016 – All rights reserved



end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.65 proofState (Spelling and Grammatical Checking State)

This element specifies if the grammar and spell checking engines of the last application to process this document completed checking the grammar and spelling of a the document before the document was last saved. Applications which modify the document contents without checking spelling or grammar should reset these states as needed.

[*Note*: If this element specifies that an application's grammar and spell checking engines completed checking the grammar and spelling of the document when the document was last saved, then subsequent applications might elect to not run their grammar and spell checking engines when the given WordprocessingML document is loaded.

This might increase the speed with which the hosting application loads the file, and does not compromise the state of the grammar or spell checking of the document, as all errors have already been found and flagged with the proofErr element (§17.13.8.1) as the document has not been edited, only loaded, since it was last saved. *end note*]

[*Example*: Consider a WordprocessingML document that is saved by a hosting application whose spelling and grammar checking engines have completed checking grammar and spelling in the given WordprocessingML document. This state is specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:proofState w:spelling="clean" w:grammar="clean" />
```

The proofState element's attributes spelling and grammar attribute both have the value `clean` specifying that the hosting application's grammar and spell checking engines completed checking both the grammar and spelling of the given document when it was last saved. *end example*]

Attributes	Description
grammar (Grammatical Checking State)	Specifies if an application's grammar checking engine completed checking the grammatical content of the document when it was last saved. If this attribute is omitted, then its value is assumed to be `dirty` (not complete). [*Example*: Consider a WordprocessingML document saved by a hosting application whose spelling and grammar checking engines have completed checking grammar and spelling in the given WordprocessingML document. This state is specified using the following WordprocessingML in the document settings: `<w:proofState w:spelling="clean" w:grammar="clean" />` The grammar attribute has the value `clean` specifying that the hosting application's grammar checking engine completed checking the grammar of the given document when it was last saved. *end example*] The possible values for this attribute are defined by the ST_Proof simple type (§17.18.69).
spelling (Spell Checking State)	Specifies if an application's spell checking engine completed checking the spelling of the document when it was last saved. If this attribute is omitted, then its value is assumed to be `dirty` (not complete). [*Example*: Consider a WordprocessingML document saved by a hosting application whose spelling and grammar checking engines have completed checking grammar and spelling in the given WordprocessingML document. This state is specified using the following WordprocessingML in the document settings: `<w:proofState w:spelling="clean" w:grammar="clean" />` The spelling attribute has the value `clean` specifying that the hosting application's spell checking engine completed checking the spelling of the given document when it was last saved. *end example*] The possible values for this attribute are defined by the ST_Proof simple type (§17.18.69).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Proof) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.66 readModeInkLockDown (Freeze Document Layout)

This element specifies the exact set of page and text sizing parameters which shall be used to display the contents of a WordprocessingML document. [*Rationale*: This setting is typically used for documents that have been annotated using ink. This setting freezes the document's presentation such that the ink annotations must exist at the same position of the WordprocessingML document irrespective of the monitor on which the WordprocessingML document is rendered. *end rationale*]

This element shall only affect the display of WordprocessingML documents as follows:

- When the actualPage attribute is specified with a value of `true`, the given WordprocessingML document's pages shall be rendered as they would normally be displayed. The resulting pages can have their magnification setting changed as desired. All other attributes shall be ignored.
- When the actualPage attribute is specified with a value of `false`, the given WordprocessingML document's pages shall be rendered as *virtual pages* when loaded by a conforming hosting application irrespective of the given WordprocessingML document's view (§17.15.1.92). *Virtual pages* are pages with no correlation with the printed layout of a given WordprocessingML document that have been scaled by a conforming hosting application to improve the readability of a given WordprocessingML document when it is displayed. Specifically, the w and h attributes specify the width and height of the virtual pages, and the fontSz attribute specifies the scaling to be applied to text within the given WordprocessingML document.

[*Example*: Consider a WordprocessingML document that must be displayed using virtual pages when its contents are displayed. This state is specified using the following WordprocessingML in the document settings:

```
<w:readModeInkLockDown w:w="692" w:h="986" w:fontSz="95" w:actualPg="0"/>
```

The readModeInkLockDown element has w and h attribute values which specify the width and height of the virtual pages to be used to render the given WordprocessingML document. Finally, the fontSz attribute specifies the scaling to be applied to text within the given WordprocessingML document. *end example*]

Attributes	Description
actualPg (Use Actual Pages, Not Virtual Pages)	Specifies if applications shall render this WordprocessingML document with actual pages, not virtual pages. A*ctual pages* are pages rendered as they is printed. A value of `true` specifies that the given WordprocessingML document's pages is rendered as they are printed, and the w, h, and fontSz attributes shall be ignored. A value of `false` specifies that the given WordprocessingML document's pages shall be rendered as virtual pages using the other attributes on this element. [*Example*: Consider a WordprocessingML document that must be displayed using virtual pages. This state is specified using the following WordprocessingML in the document settings: `<w:readModeInkLockDown w:w="692" w:h="986" w:fontSz="95" w:actualPg="0" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The actualPage attribute is equal to `0` specifying that the given WordprocessingML document must be rendered by conforming hosting applications using virtual pages. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
fontSz (Font Size Scaling)	Specifies the percentage that text in a given WordprocessingML document shall be scaled by before it is displayed on a virtual page, including a trailing percent sign (U+0025). This attribute shall only be used if the actualPage attribute equals `off`, `0`, or `false`. [*Example*: Consider a WordprocessingML document that must be displayed using virtual pages. This state is specified using the following WordprocessingML in the document settings: `<w:readModeInkLockDown w:w="692" w:h="986" w:fontSz="95%" w:actualPg="0" />` The fontSz attribute is equal to 95% specifying that the text in the WordprocessingML document must be displayed at 95% of its normal size when it is displayed on a virtual page. *end example*] The possible values for this attribute are defined by the ST_DecimalNumberOrPercent simple type (§17.18.11).
h (Virtual Page Height)	Specifies the height of the virtual pages which shall be used in this document. This value is specified in pixels. This attribute shall only be used if the actualPage attribute equals `off`, `0`, or `false`. [*Example*: Consider a WordprocessingML document that must be displayed using virtual pages. This state is specified using the following WordprocessingML in the document settings: `<w:readModeInkLockDown w:w="692" w:h="986" w:fontSz="95" w:actualPg="0" />` The h attribute is equal to 986 specifying that virtual pages in this document must be 986 pixels high. *end example*] The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§17.18.67).
w (Virtual Page Width)	Specifies the width of the virtual pages which shall be used in this document. This value is specified in pixels. This attribute shall only be used if the actualPage attribute equals `off`, `0`, or `false`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider a WordprocessingML document that must be displayed using virtual pages. This state is specified using the following WordprocessingML in the document settings: `<w:readModeInkLockDown w:w="692" w:h="986" w:fontSz="95" w:actualPg="0" />` The w attribute is equal to 692 specifying that virtual pages in this document must be 692 pixels wide. *end example*] The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§17.18.67).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ReadingModeInkLockDown) is located in §A.1. *end note*]

17.15.1.67 removeDateAndTime (Remove Date and Time from Annotations)

This element specifies that the date and time information shall be removed from all annotations which are present in the current document when it is saved. Annotations store this information in the date attribute on the annotation's XML element.

If this element is omitted, then date information shall not be removed when the document is saved. If the removePersonalInformation element is not turned on, then this setting shall be ignored.

[*Example*: Consider a WordprocessingML document that must not save date and time information on annotations in the document content. This state is specified using the following WordprocessingML in the document settings:

```
<w:settings>
  …
  <w:removePersonalInformation w:val="true" />
  <w:removeDateAndTime w:val="true" />
  …
</w:settings>
```

The removeDateAndTime element's val attribute has a value of `true` specifying that all annotations in the document must have and date and time information removed before they are saved by omitting their date attributes. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.1.68 removePersonalInformation (Remove Personal Information from Document Properties)

This element specifies that hosting applications shall remove all personal information of document authors upon saving a given WordprocessingML document. The definition and extent of personal information is not defined by ISO/IEC 29500.

If this element is omitted, then personal information shall not be removed when the document is saved.

[*Example*: Consider a WordprocessingML document that must not save personal information in the document. This state is specified using the following WordprocessingML in the document settings:

```
<w:removePersonalInformation w:val="true" />
```

The removePersonalInformation element's val attribute has a value of `true` specifying that applications must remove any personal information when saving this file. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.69 revisionView (Visibility of Annotation Types)

This element specifies which forms of annotations shall be visible for a WordprocessingML document when it is displayed. This setting shall not affect whether annotations are added or persisted, it shall only affect the display of the annotations which exist in the document's contents (persisted or in memory).

If this element is omitted, then all forms of annotations shall be visible.

[*Example*: Consider the WordprocessingML below specifying that only formatting and ink annotations within a given WordprocessingML document must be displayed when the document is opened:

```
<w:revisionView w:markup="false" w:comments="false" w:insDel="false" />
```

The revisionView element specifies that the visibility of the markup region, comments and content additions/deletions must be suppressed by setting a value of `false`. Since the formatting and inkAnnotation attributes are omitted, they inherit the default of `true` and must be displayed. *end example*]

Attributes	Description
comments (Display Comments)	Specifies if comments should be included when the contents of this document are displayed. If this attribute is omitted, then comments shall be displayed when annotations are visible based on application-level settings. [*Example*: Consider the WordprocessingML below specifying that comments must be displayed: `<w:revisionView w:comments="true" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The comments attribute has a value of `true`, specifying that comments must be rendered when the document's annotations are displayed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
formatting (Display Formatting Revisions)	Specifies if revisions to properties (i.e. formatting revisions) should be included when the contents of this document are displayed. If this attribute is omitted, then formatting revisions shall be displayed when annotations are visible based on application-level settings. [*Example*: Consider the WordprocessingML below specifying that formatting revisions must be displayed: `<w:revisionView w:formatting="true" />` The formatting attribute has a value of `true`, specifying that formatting revisions must be rendered when the document's annotations are displayed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
inkAnnotations (Display Ink Annotations)	Specifies if ink annotations should be included when the contents of this document are displayed. If this attribute is omitted, then ink annotations shall be displayed when annotations are visible based on application-level settings. [*Example*: Consider the WordprocessingML below specifying that ink annotations must be displayed: `<w:revisionView w:inkAnnotations="true" />` The inkAnnotations attribute has a value of `true`, specifying that ink annotations must be rendered when the document's annotations are displayed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
insDel (Display Content Revisions)	Specifies if revisions to content (i.e. insertions, deletions, and moves) should be included when the contents of this document are displayed. If this attribute is omitted, then insertions, deletions, and moves shall be displayed when annotations are visible based on application-level settings. [*Example*: Consider the WordprocessingML below specifying that insertions, deletions, and moves must be displayed: `<w:revisionView w:insDel="true" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The insDel attribute has a value of `true`, specifying that insertions, deletions, and moves must be rendered when the document's annotations are displayed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
markup (Display Visual Indicator Of Markup Area)	Specifies if the application shall visually indicate any additional non-printing area used to display annotations when the annotations in this document are displayed. If this attribute is omitted, then any additional non-printing area shall be indicated when they are visible based on application-level settings. [*Example*: Consider the WordprocessingML below specifying that no visual indicator must be displayed for non-printing regions holding annotations: `<w:revisionView w:markup="false" />` The markup attribute has a value of `false`, specifying that nothing must be rendered indicating when a non-printing region is added when the document's annotations are displayed. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrackChangesView) is located in §A.1. *end note*]

17.15.1.70 rsid (Single Session Revision Save ID)

This element specifies the revision save ID that was associated with a single editing session for a document. An editing session is a span of time that begins and ends with any event that produces an editable file, such as a save or an e-mail send, and contains no such event. When revision save IDs are added to a document, they shall follow these rules:

- Every editing session shall be assigned a revision save ID that is larger than all earlier ones in the same file
- Revision save IDs should be randomly generated based on the current time (to minimize the chance that two disparate editing sessions starting with the same immediate predecessor are assigned the same revision save ID)
- Changes to document content in an editing session shall be stamped with the current revision save ID using the appropriate rsid* attributes
- An identical rsid value between two documents with the same rsidRoot (§17.15.1.71) shall indicate the same editing sessions

[*Note*: A revision save ID should be treated as unique within the context of all documents with the same rsidRoot value. Although in practice it is possible for two independent sessions to result in the same value, this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

outcome is extremely rare as the values are based on the current time. However, the meaning of two revision save IDs is not defined for documents with a different rsidRoot. Applications can use this information as desired. *end note*]

[*Example*: Consider the following fragments from two WordprocessingML documents' document settings:

Document 1	Document 2
<w:rsids> <w:rsidRoot w:val="00464813"/> <w:rsid w:val="00455AAB" /> <w:rsid w:val="00464813" /> <w:rsid w:val="00996E03" /> </w:rsids>	<w:rsids> <w:rsidRoot w:val="00464813"/> <w:rsid w:val="00455AAB" /> <w:rsid w:val="00464813" /> <w:rsid w:val="00473403" /> <w:rsid w:val="0048414E" /> </w:rsids>

The rsid elements are identical for the first three editing sessions for both documents, indicating that these documents, although they are now separate, originated from the same document. The documents were then separated and the first was saved once afterwards; and the second, twice. *end example*]

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following value for an attribute of simple type ST_LongHexNumber: 00BE2C6C. This value is permitted, as it contains four hexadecimal digits, each an encoding of an octet of the actual decimal number value. It can therefore be interpreted as desired in the context of the parent XML element, *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LongHexNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.1.71 rsidRoot (Original Document Revision Save ID)

This element specifies the revision save ID which was associated with the first editing session for this document. [*Note*: This information must be identical between any number of copies of the same document, as they all originate from the same original editing session. Applications can use this information as desired. *end note*]

If this element is omitted, then the original document revision save ID is unknown.

[*Example*: Consider the following fragments from two WordprocessingML documents' document settings:

©ISO/IEC 2016 – All rights reserved

Document 1	Document 2
<w:rsids> <w:rsidRoot w:val="00464813"/> <w:rsid w:val="00455AAB" /> <w:rsid w:val="00464813" /> <w:rsid w:val="00996E03" /> </w:rsids>	<w:rsids> <w:rsidRoot w:val="00464813"/> <w:rsid w:val="00455AAB" /> <w:rsid w:val="00464813" /> <w:rsid w:val="00473403" /> <w:rsid w:val="0048414E" /> </w:rsids>

The rsidRoot element's val attribute has a value of 00464813 for both documents, indicating that these documents, although they are now separate, originated from the same document. This information can be used as desired. *end example*]

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following value for an attribute of simple type ST_LongHexNumber: 00BE2C6C. This value is permitted, as it contains four hexadecimal digits, each an encoding of an octet of the actual decimal number value. It can therefore be interpreted as desired in the context of the parent XML element, *end example*] The possible values for this attribute are defined by the ST_LongHexNumber simple type (§17.18.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LongHexNumber) is located in §A.1. *end note*]

17.15.1.72 rsids (Listing of All Revision Save ID Values)

This element specifies the set of revision save ID values for the current document. *Revision save ID values* refer to four digit hexadecimal values which uniquely identify an editing session in the life of the current document. An *editing session* is the period of time between two subsequent save operations by an application.

[*Guidance*: The set of revision save IDs stored with a document only supplies information about the editing session in which document components were last saved, which can be used by applications in any manner desired. *end guidance*]

If this element is omitted, then no information is available about the set of revision save ID values for this document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a WordprocessingML document with the following information present in its document settings:

```
<w:rsids>
  <w:rsidRoot w:val="00464813" />
  <w:rsid w:val="00455AAB" />
  <w:rsid w:val="00464813" />
  <w:rsid w:val="00473403" />
</w:rsids>
```

The rsids element contains four child elements, specifying that the document was edited over four distinct editing sessions (i.e. it was saved three times). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DocRsids) is located in §A.1. *end note*]

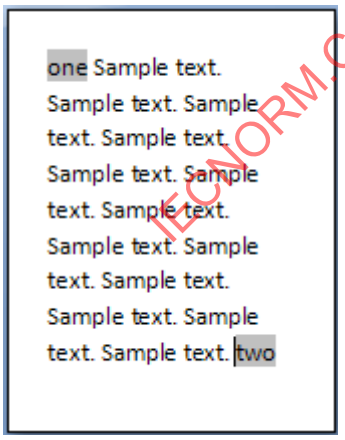
17.15.1.73 saveFormsData (Only Save Form Field Content)

This element specifies that saving the contents of this document shall only save the contents of WordprocessingML form fields defined using the FORMTEXT, FORMCHECKBOX, and FORMDROPDOWN field codes in a comma-delimited text format which does not conform to ISO/IEC 29500 (i.e. it is a one-way export from a WordprocessingML document).

[*Rationale*: This setting is typically used to allow duplication of paper forms in electronic WordprocessingML document form, allowing the resulting content to be extracted as a comma-delimited text file. *end rationale*]

If this element is omitted, then the contents of the entire document (not just form fields) should be saved according to the definition of WordprocessingML in ISO/IEC 29500.

[*Example*: Consider a WordprocessingML document which has form fields in the top right and bottom left corners of the first page, as follows (with the text box form fields shaded in grey):



If the only content which must be saved are the form fields' contents, this requirement is specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:saveFormsData w:val="true" />
```

The saveFormsData element's val attribute as a value of `true`, specifying that only form field data must be saved, resulting in output as follows in a text file:

```
one,two
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.74 saveInvalidXml (Allow Saving Document As XML File When Custom XML Markup Is Invalid)

This element specifies that this document should be capable of being saved into a format consisting of a single XML file (not defined by ISO/IEC 29500) even when its contents are invalid based XML schema validation of the custom XML markup contained in the document. This setting has no effect on documents that do not contain custom XML markup, or that do contain custom XML markup but do not have a schema attached. [*Guidance*: Because this setting specifies behavior when saving to an alternative file format not defined by ISO/IEC 29500, this behavior is optional. *end guidance*]

If this element is omitted, then applications should not allow this document to be saved into a single XML file when its contents are invalid based on the custom XML markup contained in the document. If the doNotValidateAgainstSchema element (§17.15.1.43) is set, then the XML is never "invalid" and this property is ignored.

[*Example*: Consider a WordprocessingML document which should be saved into a single XML file even when its custom XML content is marked invalid by applications which support this operation. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:saveInvalidXml w:val="true" />
```

The saveInvalidXml element's val attribute has a value of `true` specifying that the content in this document can be saved regardless of its validation status. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.75 savePreviewPicture (Generate Thumbnail For Document On Save)

This element specifies if a document's Thumbnail part should be generated for the contents of the first page of this document when saved by application which support document thumbnail generation.

If this element is omitted, then applications can choose to save a thumbnail, however, that behavior is not required. If this element is specified, a thumbnail shall be produced if that functionality is supported.

[*Example*: Consider a WordprocessingML document that specifies that a document thumbnail must always be created when it is saved. This requirement would be specified using the following WordprocessingML in the document settings part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:savePreviewPicture w:val="true"/>
```

The savePreviewPicture element's val attribute has a value of `true` specifying that a document thumbnail should be generated each time this document is saved. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.76 saveThroughXslt (Custom XSL Transform To Use When Saving As XML File)

This element specifies the location of a custom XSL transform that shall be used when this document is saved as a single XML file (in an application-defined format). [*Guidance*: Because this setting specifies behavior when saving to an alternative file format not defined by ISO/IEC 29500, this behavior is optional. *end guidance*]

If this element is omitted, then no custom XSL transform shall be used when saving this file as a single XML file. If the useXSLTWhenSaving element (§17.15.1.91) is omitted or set to `false`, then this transform shall not be applied when the document is saved as a single XML file.

[*Example*: Consider a XML document that must have the XSL transform applied when the document is saved as a single XML file. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:useXSLTWhenSaving w:val="on"/>
<w:saveThroughXslt r:id="rId5" />
```

The useXSLTWhenSaving element's val is set to `true` indicating that applications must apply the XSLT specified by the relationship targeted by the id attribute of the saveThroughXslt element, located at `rId5`, when saving as a single XML file. *end example*]

Attributes	Description
id (XSL Transformation Location) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies an explicit relationship to the location of the XSL Transformation which shall be applied. The relationship targeted by this element shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/transform, or this document shall be declared non-conformant. If neither this attribute nor the solutionID attribute are present, no XSLT transformation shall be performed. [*Example*: Consider a XML document that must have the XSL transform located at `c:\Example Transform.xslt` applied when the document is saved as a single XML file. This requirement would be specified using the following WordprocessingML in the document settings: `<w:saveThroughXslt r:id="rId5" />` The saveThroughXslt element specifies that the relationship located at `rId5` must be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	used when saving as a single XML file in this case, that relationship must target `c:\Example Transform.xslt`. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
solutionID (Local Identifier for XSL Transform)	Specifies a string identifier that can be used to locate the XSL transform to be applied. The semantics of this attribute are application-defined - applications can use this information in any application-defined manner to resolve the location of the XSL transform to apply. If this attribute is omitted, then no local identifier is specified for the XSL transform. If both this and the xslt attributes are present, then this data shall be used first, and the latter shall only be used if this information cannot be used successfully. [*Example*: Consider a XML document that must have the XSL transform identified by `mySolution` applied to when the document is saved as a single XML file. This requirement would be specified using the following WordprocessingML in the document settings: `<w:saveThroughXslt w:solutionID="mySolution" />` The solutionID attribute has a value of `mySolution` indicating that applications must apply the XSLT identified by this value (if known) when saving as a single XML file. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SaveThroughXslt) is located in §A.1. *end note*]

17.15.1.77 saveXmlDataOnly (Only Save Custom XML Markup)

This element specifies that the contents of this document shall be saved as an XML file containing only the custom XML markup in this document in its regular form. The resulting document does not conform to ISO/IEC 29500 (i.e. this is an export-only save option for a WordprocessingML document).

[*Rationale*: This setting is typically used to extract custom XML markup from a WordprocessingML document for further processing by XML-enabled applications. *end rationale*]

If this element is omitted, then the contents of the entire document (not just custom XML markup) should be saved according to the definition of WordprocessingML in ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a WordprocessingML document which should be saved as an XML file containing only its custom XML markup. This requirement is specified using the following WordprocessingML fragment in the document settings:

```
<w:saveXmlDataOnly w:val="true"/>
```

The saveXmlDataOnly element's val attribute has a value of `true` specifying that only custom XML must be saved into a regular XML file when saving this document. For example, the document body formerly looked like this:

```
<w:body>
  <w:p>
    <w:customXml w:element="root" w:uri="urn:example">
      <w:r>
        <w:t>Hello, world</w:t>
      </w:r>
    </w:customXml>
  </w:p>
</w:body>
```

The presence of this element specifies that the resulting document only contains the custom Xml markup, resulting in the following:

```
<ns0:root xmlns:ns0="urn:example">Hello, world</ns0:root>
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.78 settings (Document Settings)

This element specifies the settings that are applied to a WordprocessingML document. This element is the root element of the Document Settings part in a WordprocessingML document.

[*Example*: Consider the following WordprocessingML fragment for the settings part of a document:

```
<w:settings>
  <w:defaultTabStop w:val="720" />
  <w:characterSpacingControl w:val="doNotCompress" />
</w:settings>
```

The settings element contains all of the settings for this document. In this case, the two settings applied are automatic tab stop increments of 0.5" using the defaultTabStop element, and no character level whitespace compression using the characterSpacingControl element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Settings) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.79 showEnvelope (Show E-Mail Message Header)

This element specifies that an e-mail message header shall be displayed when this document is opened, if an e-mail header is supported by the application opening the file.

If this element is omitted, then applications shall not display the e-mail message header automatically when this file is opened, even if one is available in the application opening the file.

[*Example*: Consider a WordprocessingML document which should show an e-mail message header when opened. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:showEnvelope w:val="true" />
```

The showEnvelope element's val attribute has a value of `true` specifying that an e-mail message header must be displayed when the document is viewed, whenever such functionality is available. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.80 showXMLTags (Show Visual Indicators for Custom XML Markup Start/End Locations)

This element specifies that some visual indicator shall be provided for the start and end locations of custom XML markup present in this document, if any.

If this element is omitted, then applications should not provide any visual indicator of the locations of custom XML markup start/end tags.

[*Example*: Consider a WordprocessingML document which should show a visual indicator to the location of custom XML markup elements. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:showXMLTags w:val="true" />
```

The showXMLTags element's val attribute has a value of `true` specifying that custom XML markup should have a visual indicator in the document when displayed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.81 smartTagType (Supplementary Smart Tag Information)

This element specifies optional supplementary information about one or more smart tags (§17.5.1.9) used in the current WordprocessingML document. This supplementary data is linked to the smart tag to which it applies via its name and namespaceuri attributes.

[*Example*: Consider a smart tag which has supplementary information defined as using the following WordprocessingML:

```
<w:smartTagType w:name="companyName" w:namespaceuri="urn:smartTagExample"
w:url="http://www.example.com/smartTag"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The name and namespaceuri attributes specify that the smart tag to which this data must be companyName in the urn:smartTagExample namespace. The supplementary data is an associated URL of http://www.example.com/smartTag. *end example*]

Attributes	Description
name (Smart Tag Name)	Specifies the name of the smart tag within the document for which supplementary data is provided. [*Example*: Consider a smart tag which has a name of companyName. This name would be referenced using the following WordprocessingML: `<w:smartTagType w:name="companyName" … />` The name attribute specifies that the name for this smart tag must be companyName. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
namespaceuri (Smart Tag Namespace)	Specifies the namespace URI of the smart tag for which supplementary data is provided. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). [*Example*: Consider a smart tag which must have a namespace URI of urn:smartTagExample. This namespace would be referenced using the following WordprocessingML: `<w:smartTagType w:namespaceuri="urn:smartTagExample" />` The namespaceuri attribute specifies that the namespace for the smart tag to which this data applies must be urn:smartTagExample. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
url (Smart Tag Supplementary URL)	Specifies a URL provided for a particular smart tag type in this document. [*Note*: This URL is typically used to provide access to a URL for additional updates to this smart tag type as requested by the smart tag provider. *end note*] If this attribute is omitted, then no supplementary URL is provided for this type. [*Example*: Consider a smart tag which must have a supplementary URL of http://www.example.com/smartTag. This URL would be specified using the following WordprocessingML: `<w:smartTagType … w:url="http://www.example.com/smartTag" />` The url attribute specifies that the supplementary data for the smart tag to which this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	data applies must be `http://www.example.com/smartTag`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagType) is located in §A.1. *end note*]

17.15.1.82 strictFirstAndLastChars (Use Strict Kinsoku Rules for Japanese Text)

This element specifies that the strict set of Kinsoku rules shall be applied to Japanese text in this document when the kinsoku element (§17.3.1.16) is applied to that text. The resulting line breaking rules are provided on the kinsoku element.

If this element is omitted, then standard rules shall apply to Japanese text when the kinsoku element is applied to that text.

[*Example*: Consider a WordprocessingML document that specifies that strict Kinsoku rules must be applied to Japanese text. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:strictFirstAndLastChars w:val="true"/>
```

The strictFirstAndLastChars element's val attribute has a value of `true` specifying that a document must apply the strict set of disallowed characters for the start and end of a line. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.83 styleLockQFSet (Prevent Replacement of Styles Part)

This element specifies whether applications shall prevent the replacement of the complete set of styles stored in the Styles part when editing this document. This setting should not preclude the editing or removal of individual styles, instead, it should only prevent the removal and replacement of the entire styles part in a single operation (either through a user interface or a programmatic operation).

If this element is omitted, then applications can allow the replacement of the entire styles part in this document.

[*Example*: Consider a WordprocessingML document that specifies that applications must prevent the replacement of the entire styles part. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:styleLockQFSet w:val="true"/>
```

The styleLockQFSet element's val attribute has a value of `true` specifying that individual style changes should be allowed, but the styles data must not be replaced as a whole via a single operation. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.84 styleLockTheme (Prevent Modification of Themes Part)

This element specifies whether applications shall prevent the modification of the document's theme information stored in the Theme part when editing this document. This setting should not preclude the use of the theme information, instead, it should only prevent the modification of the theme part in a single operation (either through a user interface or a programmatic operation).

If this element is omitted, then applications can allow the replacement or modification of the theme part in this document.

[*Example*: Consider a WordprocessingML document that specifies that applications must prevent the modification of the theme part. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:styleLockTheme w:val="true"/>
```

The styleLockTheme element's val attribute has a value of `true` specifying that theme data must not be modified when modifying the contents of this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.85 stylePaneFormatFilter (Suggested Filtering for List of Document Styles)

This element specifies a set of suggested filters which should be applied to the list of document styles in this application if the styles are displayed in a user interface.

If this element is omitted, then all settings defined by this element are turned off.

[*Example*: Consider a document with the following value in its document settings:

```
<w:stylePaneFormatFilter w:customStyles="true" w:top3HeadingStyles="true" />
```

The stylePaneFormatFilter element's settings specify two suggested filter options for the list of document styles:

- Only custom styles should be shown
- Heading styles with a styleId of `Heading1` to `Heading3` should always be displayed in the list

end example]

Attributes	Description
allStyles (Display All Styles)	Specifies that all styles present in the Styles part should be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
alternateStyleNam	Specifies that primary names for styles should not be shown if an alternate name using

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
es (Use the Alternate Style Name)	the name element (§17.7.4.9) exists. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
clearFormatting (Display Styles to Remove Formatting)	Specifies that a style should be present which removes all formatting and styles from text. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
customStyles (Display Only Custom Styles)	Specifies that only styles with the customStyle attribute should be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
directFormattingOnNumbering (Display Direct Formatting on Numbering Data)	Specifies that all unique forms of direct formatting of numbering data should be displayed in the list of document styles as though they were each a unique style. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
directFormattingOnParagraphs (Display Paragraph Level Direct Formatting)	Specifies that all unique forms of paragraph-level direct formatting should be displayed in the list of document styles as though they were each a unique style. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
directFormattingOnRuns (Display Run Level Direct Formatting)	Specifies that all unique forms of run-level direct formatting should be displayed in the list of document styles as though they were each a unique style. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
directFormattingOnTables (Display Direct Formatting on Tables)	Specifies that all unique forms of direct formatting of tables should be displayed in the list of document styles as though they were each a unique style. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
headingStyles (Display Heading Styles)	Specifies that heading styles (styles with a styleId of `Heading1` to `Heading9`) should be displayed in the list of document styles when the previous style is used in the document and/or is present in the Styles part. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
latentStyles (Display Latent Styles)	Specifies that all latent styles should be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
numberingStyles (Display Numbering Styles)	Specifies that numbering styles should be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
stylesInUse (Display Styles in Use)	Specifies that only styles used in the document should be displayed in the list of document styles.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
tableStyles (Display Table Styles)	Specifies that table styles should be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
top3HeadingStyles (Display Heading 1 through 3)	Specifies that heading styles with a styleId of `Heading1` to `Heading3` should always be displayed in the list of document styles. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
visibleStyles (Only Show Visible Styles)	Specifies that styles should only be shown if the semiHidden element (§17.7.4.16) is false and the hidden element (§17.7.4.4) is false. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StylePaneFilter) is located in §A.1. *end note*]

17.15.1.86 stylePaneSortMethod (Suggested Sorting for List of Document Styles)

This element specifies a sorting which should be applied to the list of styles in this document if the styles are displayed in a user interface.

If this element is omitted, then styles which are visible should be sorted by the default sorting of the host application.

[*Example*: Consider a document with the following value in its document settings:

```
<w:stylePaneSortMethod w:val="type" />
```

The stylePaneFormatFilter element's val attribute specifies that styles which are visible should be sorted by their style types (i.e. character, linked, paragraph) via a value of `type`. *end example*]

Attributes	Description
val (Style Sorting)	Specifies a sort order which should be applied to the list of document styles when they are displayed in a user interface. The possible values for this attribute are defined by the ST_StyleSort simple type (§17.18.82).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleSort) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.1.87 summaryLength (Percentage of Document to Use When Generating Summary)

This element specifies the size for automatic document summaries performed on the content of a WordprocessingML document. An *automatic document summary* is a subset of text contained in a document deemed by the hosting application to summarize the content of the WordprocessingML document. The val attribute of this element specifies the size of an automatic document summary to be performed on a given WordprocessingML document as a percentage of the total size of the given WordprocessingML document. Performing an automatic document summary is a runtime operation outside the scope of ISO/IEC 29500.

If this element is omitted, then applications can summarize this document to any desired size.

[*Example*: Consider a WordprocessingML document whose automatic document summary is to be ten percent of the size of the given WordprocessingML document. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:summaryLength w:val="10%" />
```

The summaryLength element's val attribute is equal to 10% specifying that any automatic document summary is to be ten percent of the size of the document. *end example*]

Attributes	Description
val (Value in Percent)	Specifies a measurement in whole percentage points, with a trailing percent sign (U+0025). [*Example*: Consider the following WordprocessingML fragment: `<… w:val="50.125%" />` This value specifies a value of fifty and one-eighth percent. *end example*] The possible values for this attribute are defined by the ST_DecimalNumberOrPercent simple type (§17.18.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumberOrPrecent) is located in §A.1. *end note*]

17.15.1.88 themeFontLang (Theme Font Languages)

This element specifies the language which shall be used to determine the appropriate theme fonts in the document's Theme part which map to the major/minor theme fonts. Specifically, the bidi attribute is used to determine the theme font applied to complex script text, the eastAsia attribute is used to determine the theme font applied to East Asian text, and the val attribute is used to determine the theme font applied to all other text.

These mappings are performed as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- For `majorAscii`/`majorHAnsi`, locate the font element (§20.1.4.1.16) in the majorFont element (§20.1.4.1.24) in the theme part for the language specified by the val attribute
- For `majorBidi`, locate the font element in the majorFont element in the theme part for the language specified by the bidi attribute
- For `majorEastAsia`, locate the font element in the majorFont element in the theme part for the language specified by the eastAsia attribute
- For `minorAscii`/`minorHAnsi`, locate the font element in the minorFont element (§20.1.4.1.25) in the theme part for the language specified by the val attribute
- For `minorBidi`, locate the font element in the minorFont element in the theme part for the language specified by the bidi attribute
- For `minorEastAsia`, locate the font element in the minorFont element in the theme part for the language specified by the eastAsia attribute

If this element is omitted, then the default fonts for each region as specified by the latin, ea, and cs elements (§21.1.2.3.7; §21.1.2.3.3; §21.1.2.3.1) should be used.

[*Example*: Consider a document with the following WordprocessingML in its document settings:

```
<w:themeFontLang w:val="ja-JP" />
```

The themeFontLang element's val attribute has a value of `ja-JP`, specifying that the theme fonts used for Latin text must be the theme fonts for Japanese. If the following content was present in the theme part:

```
…
<a:majorFont>
  …
  <a:font script="Jpan" typeface="MS Mincho"/>
  …
</a:majorFont>
…
```

Then this setting would specify that uses of the `majorAscii` and `majorHAnsi` theme font enumerations must be mapped to the `MS Mincho` font. *end example*]

Attributes	Description
bidi (Complex Script Language)	Specifies the language which shall be used when processing the contents of this run which use complex script characters, as determined by the Unicode character values of the run content. If this attribute is omitted, then the languages for the contents of this run using complex script characters shall be automatically determined based on their contents using any appropriate method. [*Example*: Consider a run which contains complex script characters in its contents. If those contents should be interpreted as Hebrew, that requirement would be specified as

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	follows in the resulting WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:lang w:bidi="he-IL" />`
	`  </w:rPr>`
	`</w:r>`
	
	The resulting run specifies that any complex script contents must be spell and grammar checked using a Hebrew dictionary and grammar engine, if one is available. *end example*]
	
	The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

eastAsia (East Asian Language)	Specifies the language which shall be used when processing the contents of this run which use East Asian characters, as determined by the Unicode character values of the run content.
	
	If this attribute is omitted, then the languages for the contents of this run using East Asian characters shall be automatically determined based on their contents using any appropriate method.
	
	[*Example*: Consider a run which contains East Asian characters in its contents. If those contents should be interpreted as Korean, that requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:r>`
	`  <w:rPr>`
	`  <w:lang w:eastAsia="ko-KR" />`
	`  </w:rPr>`
	`</w:r>`
	
	The resulting run specifies that any complex script contents must be spell and grammar checked using a Korean dictionary and grammar engine, if one is available. *end example*]
	
	The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

val (Latin Language)	Specifies the language which shall be used to check spelling and grammar (if requested) when processing the contents of this run which use Latin characters, as determined by the Unicode character values of the run content. If this attribute is omitted, then the languages for the contents of this run using Latin characters shall be automatically determined based on their contents using any appropriate method. [*Example*: Consider a run which contains Latin characters in its contents. If those contents should be interpreted as English (Canada), that requirement would be specified as follows in the resulting WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:r> <w:rPr> <w:lang w:val="en-CA" /> </w:rPr> </w:r> The resulting run specifies that any complex script contents must be spell and grammar checked using a English (Canada) dictionary and grammar engine, if one is available. *end example*] The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Language) is located in §A.1. *end note*]

17.15.1.89 trackRevisions (Track Revisions to Document)

This element specifies that applications shall track revisions made to the WordprocessingML document. *Revisions* are changes to a WordprocessingML document which are recorded such that they can be viewed independently, accepted or removed, and reverted if needed. When revisions are tracked, the resulting WordprocessingML markup in the Revisions subclause of this document describes the necessary syntax.

If this element is omitted, then revisions shall not be generated by changes to the contents of this document.

[*Example*: Consider a WordprocessingML document containing the text run `Example` that must not have revisions tracked. Example WordprocessingML from Document 1 is given below:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

And the corresponding document settings:

```
<w:settings>
  <w:trackRevisions w:val="false"/>
  …
</w:settings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If the word `text` was added to the end of this document and bolded without revisions tracked, the resulting WordprocessingML would be output as follows:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
      <w:r>
        <w:rPr>
          <w:b/>
        </w:rPr>
        <w:t>text</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

And the corresponding document settings:

```
<w:settings>
  <w:trackRevisions w:val="false"/>
  …
</w:settings>
```

Finally, assume the same insertion and formatting took place when the trackRevisions element's val attribute was set to `true`, the resulting WordprocessingML would be output as follows:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example</w:t>
      </w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <w:ins … >
          <w:r>
            <w:rPr>
              <w:b/>
              <w:rPrChange … >
                <w:rPr/>
              </w:rPrChange>
            </w:rPr>
            <w:t>text</w:t>
          </w:r>
        </w:ins>
      </w:p>
    </w:body>
  </w:document>
```

And the corresponding document settings:

```
  <w:settings>
    <w:trackRevisions w:val="true"/>
    …
  </w:settings>
```

The trackRevisions element's val attribute was set to `true`, therefore the changes to the content of the document were inserted using the appropriate annotation elements in the document's WordprocessingML. Specifically, inserting the text `Text` to the right of the existing text was tracked as a revision with the ins element. In addition, applying bold formatting to the text was tracked as a revision with the rPrChange element. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.90 updateFields (Automatically Recalculate Fields on Open)

This element specifies whether the fields contained in this document should automatically have their field result recalculated from the field codes when this document is opened by an application which supports field calculations. [*Note*: Some fields are always recalculated (e.g. the page numbering), therefore this element only affects fields which are typically not automatically recalculated on opening the document. Also note that this setting must not supersede any document protection (§17.15.1.29) or write protection (§17.15.1.93) settings. *end note*]

If this element is omitted, then fields should not automatically be recalculated on opening this document.

[*Example*: Consider a WordprocessingML document that specifies that applications should attempt to automatically recalculate fields from their field codes upon opening this document. This requirement would be specified using the following WordprocessingML in the document settings part:

```
  <w:updateFields w:val="true"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The updateFields element's val attribute has a value of `true` specifying that all fields should automatically be recalculated when opening this document. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.91 useXSLTWhenSaving (Save Document as XML File through Custom XSL Transform)

This element specifies that this document should be saved through the custom XSLT transform defined by the saveThroughXslt element (§17.15.1.76) in this document when it is saved as a single XML file (not defined by ISO/IEC 29500). [*Guidance*: Because this setting specifies behavior when saving to an alternative file format not defined by ISO/IEC 29500, this behavior is optional. *end guidance*]

If the saveXmlDataOnly element (§17.15.1.77) is specified, then the single XML file to be transformed is the custom XML markup of the document, otherwise, it is in an implementation-defined format. If the XSL transform specified by the saveThroughXslt element is not present, then this setting should be ignored.

If this element is omitted, then this document should not be saved through a custom XSL transform when it is saved as a single XML file.

[*Example*: Consider a WordprocessingML document which should be saved through a custom XSL transform when it is saved as a single XML file. This requirement is specified using the following WordprocessingML in the document settings:

```
<w:useXSLTWhenSaving w:val="true" />
```

The useXSLTWhenSaving element's val attribute has a value of `true` specifying that the content in this document should be saved as a single XML file through the custom XSLT specified by the saveThroughXslt element. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.1.92 view (Document View Setting)

This element specifies the manner in which the contents of this document should be displayed when opened by an application. [*Note*: Although this Standard is for a file format, occasionally, guidance is given regarding intent in dealing with things outside that file format, such as the rendering of documents to a screen or printer. *end note*]

If this element is omitted, then an application can view the document in any desired default state.

Attributes	Description
val (Document View Setting Value)	Specifies the view that shall be used to render the contents of a WordprocessingML document. Applications can omit support for one or more of the views defined by the ST_View simple type (referenced below). If a WordprocessingML document containing an unsupported view is loaded by an application, it shall fall back to its default view

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§17.18.102). [*Example*: Consider a WordprocessingML document that is to be rendered in a view meant to mimic how the document would look in a web browser (i.e. without a fixed page width). This intent is specified using the following WordprocessingML in the document settings: `<w:view w:val="web" />` *end example*] The possible values for this attribute are defined by the ST_View simple type (§17.18.102).

[*Note*: The W3C XML Schema definition of this element's content model (CT_View) is located in §A.1. *end note*]

17.15.1.93 writeProtection (Write Protection)

This element specifies the write protection settings which have been applied to a WordprocessingML document. *Write protection* refers to a mode in which the document's contents cannot be edited, and the document cannot be resaved using the same file name. This setting is independent of the documentProtection (§17.15.1.29) element, but like document protection, this setting is not intended as a security feature and can be ignored.

When present, the write protection shall result in one of two write protection behaviors:

- If the password attribute is present, or both attributes are omitted, then the application shall prompt for a password to exit write protection. If the supplied password does not match the hash value in this attribute, then write protection shall be enabled.
- If only the recommended attribute is present, the application should provide user interface recommending that the user open this document in write protected state. If the user chooses to do so, the document shall be write protected, otherwise, it shall be opened fully editable.

If this element is omitted, then no write protection shall be applied to the current document.

[*Example*: Consider a WordprocessingML document that can be opened but only in a write protected state unless a password is provided, in which case the file would be opened in an editable state. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:writeProtection w:hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />
```

If the attributes specified in the password attribute group AG_Password are present, then the application shall require a password to exit write protection. If the hash value derived from supplied password does not match the hash value in the attribute hashValue, then write protection shall be enabled. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>algorithmName (Cryptographic Algorithm Name)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value.

The following values are reserved:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>MD2</td><td>Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `< … algorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The algorithmName attribute value of "SHA-1" specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
hashValue (Password Hash Value)	Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced. If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… AlgorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The hashValue attribute value of 9oN7nWkCAyEZib1RomSJTjmPpCY= specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of SHA-1) and that the resulting has value must be 9oN7nWkCAyEZib1RomSJTjmPpCY= for the protection to be disabled. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
recommended (Recommend Write Protection in User Interface)	Specifies that applications should provide user interface recommending that the user open this document in write protected state. If the user chooses to do so, the document shall be write protected, otherwise, it shall be opened fully editable. If this attribute is omitted, then user interface recommending that the user open this document in write protected state should not be provided. If the password attribute is also specified, then this setting shall be ignored. [*Example*: Consider a WordprocessingML document which specifies that applications must recommend write protection to this document. This requirement would be specified using the following WordprocessingML in the document settings:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:writeProtection w:recommended="true" /> The recommended attribute has a value of true specifying that the applications must hash any password provided, and if it matches this hash value, can only then halt enforcement of write protection. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: <… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The saltValue attribute value of ZUdHa+D8F/OAKP3I7ssUnQ== specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: <… spinCount="100000"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The spinCount attribute value of 100000 specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hashValue attribute. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WriteProtection) is located in §A.1. *end note*]

17.15.1.94 zoom (Magnification Setting)

This element specifies the magnification level which should be applied to a document when it is displayed by an application. The zoom level is specified with the use of two attributes stored on this element:

- val, which stores the type of zoom applied to the document
- percent, which stores the zoom percentage to be used when rendering the document

If both attributes are present, then the percent attribute shall be treated as a 'cached' value and only used when the value none is specified for the val attribute.

If this element is omitted, then applications can display the document in any desired magnification setting.

[*Example*: Consider a WordprocessingML document that is to have its zoom level at seventy one percent when it is displayed. This requirement would be specified using the following WordprocessingML fragment in the document settings:

```
<w:zoom w:percent="71%" />
```

The zoom element's percent attribute has a value of 71%, specifying that the given document must have its zoom level set to seventy one percent when it is displayed. *end example*]

Attributes	Description
percent (Zoom Percentage)	Specifies the zoom percentage that should be applied when a given WordprocessingML document is rendered by conforming hosting applications. This value is the zoom percentage specified as an integer with a trailing percent sign (U+0025). If this attribute is omitted, then applications can use any desired default percentage for the magnification. If the val attribute instantiated in addition to the percent attribute, then the percent attribute shall be treated as a cached value and only used when the value none is specified for the val attribute. If the value specified exceeds the maximum zoom level

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	available in a conforming hosting application, the conforming hosting application shall display the document using its maximum zoom level. Correspondingly, if the value specified is less than the minimum zoom level available in the conforming hosting application, the conforming hosting application shall display the document using its minimum zoom level. [*Example*: Consider a WordprocessingML document that is to have its zoom level at fifty percent when rendered by conforming hosting applications. This requirement would be specified using the following WordprocessingML: `<w:zoom w:percent="50%" />` The percent attribute has a value of `50%`, specifying that the given WordprocessingML document must to have its zoom level set to fifty percent when it is displayed. *end example*] The possible values for this attribute are defined by the ST_DecimalNumberOrPercent simple type (§17.18.11).
val (Zoom Type)	Specifies the type of zoom which shall be applied to a given document on open. If this attribute is not present, then the document shall be displayed as though the value had been set to none, and should rely on the value of the percent attribute for the actual zoom percentage. [*Example*: Consider a WordprocessingML document that should be visible without any horizontal scrolling when it is displayed. This requirement would be specified using the following WordprocessingML: `<w:zoom w:val="bestFit" w:percent="90%" />` The val attribute is equal to the value `bestFit` specifying that an application must dynamically calculate the magnification needed such that the given document must be visible on the horizontal plane of the document with no horizontal scrolling required to see any part of the WordprocessingML document's pages. Since both attributes are present, the percent attribute must be treated as a 'cached' value and ignored. *end example*] The possible values for this attribute are defined by the ST_Zoom simple type (§17.18.105).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Zoom) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.2 Web Page Settings

The next group of settings stored in WordprocessingML is web page settings. These settings specify two categories of settings:

- Settings which are related to HTML documents (i.e. frameset definitions) that can be used in WordprocessingML documents as well
- All settings which affect how this document shall be handled when it is saved as HTML. Actually saving a document as HTML is outside of the scope of ISO/IEC 29500, but in order to ensure the maximum interoperability between a WordprocessingML document and an HTML document, settings not explicitly stored elsewhere are stored in these settings.

[*Example*: Consider the following WordprocessingML fragment for the web page settings in a WordprocessingML document:

```
<w:webSettings>
  <w:frameset>
    …
  </w:frameset>
  <w:doNotUseLongFileNames w:val="true" />
</w:webSettings>
```

The webSettings element contains all of the web page settings for this document. In this case, the web page settings specified for this document are: a frameset defined using the frameset element (§17.15.2.19); and a setting specifying that when this file is saved as a web page, all resulting files must not exceed 8 octets with 3 octet extension using the doNotUseLongFileNames element (§17.15.2.13). *end example*]

17.15.2.1 allowPNG (Allow PNG as Graphic Format)

This element specifies that applications shall allow use of the PNG file format when the contents of this WordprocessingML document are saved as a web page. This includes all supporting images used as part of this HTML web page.

If this element is omitted from the document, then the PNG file format shall not be allowed when this document is saved as a web page, and that another suitable file format (such as the JPEG file format) should be utilized in its place.

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of PNG images. However, although PNG utilizes a lossless compression algorithm, JPEG uses 'lossy' compression and can in some cases result in lower fidelity images. *end note*]

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:webSettings>
  <w:allowPNG w:val="true" />
</w:webSettings>
```

The allowPNG element has a val attribute value of true, which specifies that applications can use the PNG graphic format as needed when saving this WordprocessingML document as a web page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.2 blockQuote (Data for HTML blockquote Element)

This element specifies that the current div element does not represent an HTML div element, but rather represents an HTML blockquote element. This element shall specify that this container shall be written out using the blockquote element if this document is subsequently saved as HTML.

If this element is omitted, then the current div element does not represent an HTML blockquote element. If both this element and the bodyDiv element (§17.15.2.3) are specified, then this element shall take precedence in all cases.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body style="margin-left:200px;margin-top:50px">
    <p>Paragraph one.</p>
    <blockquote style="border: 5px solid #00FFFF">
      <p>Paragraph in a blockquote.</p>
    </blockquote>
    <p>Paragraph two.</p>
  </body>
</html>
```

When this document is saved in the WordprocessingML format, the information stored on the div, blockquote, and body elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1626542603">
    …
    <w:divsChild>
      <w:div w:id="313534916">
        <w:blockQuote w:val="true" />
        …
      </w:div>
    </w:divsChild>
  </w:div>
</w:divs>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The blockQuote element has a val attribute value of `true`, which specifies that the nested div element actually represents a nested HTML `blockquote` when this document is resaved as HTML. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.3 bodyDiv (Data for HTML body Element)

This element specifies that the current div element does not represent an HTML `div` element, but rather represents formatting properties on the HTML body element. This element shall specify that the properties specified by this container shall be written out onto the body element if this document is subsequently saved as HTML.

If this element is omitted, then the current div element does not represent an HTML body element. If both this element and the blockQuote element (§17.15.2.2) are specified, then this element shall be ignored. If this element is specified on any div which is not the main div element for the document, then this element shall be ignored.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body style="margin-left:200px;margin-top:50px">
    <p>Paragraph one.</p>
    <blockquote style="border: 5px solid #00FFFF">
      <p>Paragraph in a blockquote.</p>
    </blockquote>
    <p>Paragraph two.</p>
  </body>
</html>
```

When this document is saved in the WordprocessingML format, the information stored on the `div`, `blockquote`, and body elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1626542603">
    <w:bodyDiv w:val="true" />
    …
    <w:divsChild>
      …
    </w:divsChild>
  </w:div>
</w:divs>
```

The bodyDiv element has a val attribute value of `true`, which specifies that the div element actually represents properties on the HTML body when this document is resaved as HTML. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.2.4 bottom (Bottom Border for HTML div)

This element specifies the border which shall be displayed at the bottom of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a bottom border.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style=" border-left-style:solid; border-right-style:groove; border-
right-width:1px; border-top-style:dashed; border-top-width:3px; border-bottom-
style:outset; border-bottom-width:3px">
      <p>paragraph of text</p>
    </div>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    …
    <w:divBdr>
      <w:top w:val="dashed" w:sz="18" w:space="7" w:color="auto" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="auto" />
      <w:bottom w:val="outset" w:sz="18" w:color="auto" />
      <w:right w:val="threeDEngrave" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The bottom element specifies border information about the bottom border for the single HTML `div` structure in the document; in this case, a 2.25 point bottom border of type `outset`. The initial 3 pixel border was converted to 2.25 points using the following logic:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

$$3\text{px} * \frac{1 \text{ inch}}{96 \text{ px}} * \frac{576 \text{ eighth points}}{1 \text{ inch}} = 18 \text{ eighth points } (2.25 \text{ points})$$

end example]

This element's content model is defined by the common border properties definition in §17.3.4.

17.15.2.5 color (Frameset Splitter Color)

This element specifies the color of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and can be ignored for all nested framesets in this document.

If this element is omitted, then the default color of the splitter can be automatically determined by the application displaying this WordprocessingML document (equivalent to a val attribute value of auto).

[*Example*: Consider a frameset consisting of the following three frames:

The following properties define the presentation of the splitter bars within this frameset:

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="200" />
    <w:color w:val="0000FF" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The color element's val attribute specifies that the splitters must be displayed in the RGB color `0000FF` (blue) when the contents of this document are displayed. *end example*]

Attributes	Description

themeColor (Run Content Theme Color)	Specifies a theme color which should be applied to the current run.
	
	The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows for color information to be set centrally in the document.
	
	If the themeColor attribute is specified, then the val attribute is ignored for this run.
	
	[*Example*: Consider a run of text which should be displayed using the `accent3` theme color from the document's Theme part. This requirement would be specified as follows in the resulting WordprocessingML:
	
	`<w:rPr>`
	`  <w:color w:themeColor="accent3" />`
	`</w:rPr>`
	
	The color attribute specifies that the run must use the `accent3` theme color. *end example*]
	
	The possible values for this attribute are defined by the ST_ThemeColor simple type (§17.18.97).

themeShade (Run Content Theme Color Shade)	Specifies the shade value applied to the supplied theme color (if any) for this run's contents. If the themeTint is supplied, the value of this attribute shall be ignored. If the themeShade is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run. The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border. [*Example*: Consider a shade of 40% applied to a run in a document. This shade is calculated as follows: $$S_{xml} = 0.4 * 255 = 102 = 66(hex)$$ The resulting themeShade value in the file format would be 66. *end example*] Given an input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a shade value S (from 0-100), the shade is applied as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	$$C' = \left(1 - \frac{S}{100}\right) \mathrm{C}$$ [*Example*: Consider a document with a run using the `accent6` theme color, whose RGB value (in RRGGBB hex format) is `F79646`. The hex value for the green component is `96` - 150 in decimal. Applying the shade formula with shade of 50%, the output decimal value of the green component is 75, or a hex value of 4B. This transformed value can be seen in the resulting run color WordprocessingML's `val` attribute: `<w:color w:val="7B4B23" w:themeColor="accent6" w:themeShade="80" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).
themeTint (Run Content Theme Color Tint)	Specifies the tint value applied to the supplied theme color (if any) for this run's contents. If the themeTint is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run. The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border. [*Example*: Consider a tint of 60% applied to a run in a document. This tint is calculated as follows: $$\begin{aligned} T_{xml} &= 0.6 * 255 \\ &= 153 \\ &= 99(hex) \end{aligned}$$ The resulting themeTint value in the file format would be `99`. *end example*] Given an input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a tint value T (from 0-100), the tint is applied as follows: $$C' = \left(1 - \frac{T}{100}\right)(255 - \mathrm{C}) + \mathrm{C}$$ [*Example*: Consider a document with a run using the accent1 theme color, whose RGB value (in RRGGBB hex format) is `C0504D`. The hex value for the green component is `50` - 80 in decimal. Applying the tint formula

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	with tint of 60%, the output decimal value of the green component is 150, or a hex value of 96. This transformed value can be seen in the resulting run color's WordprocessingML val attribute: `<w:color w:val="D99694" w:themeColor="accent1" w:themeTint="99" />` *end example*] The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§17.18.98).

val (Run Content Color)	Specifies the color for this run.
	
	This color can either be presented as a hex value (in RRGGBB format), or `auto` to allow a consumer to automatically determine the run color as appropriate.
	
	If the run specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value.
	
	[*Example*: Consider a run color with value `auto`, as follows:
	
	`<w:rPr>`
	`<w:color … w:val="auto" />`
	`</w:rPr>`
	
	This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the run contents can be distinguished against the page's background color. *end example*]
	
	The possible values for this attribute are defined by the ST_HexColor simple type (§17.18.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.1. *end note*]

17.15.2.6 div (Information About Single HTML div Element)

This element specifies information about a single HTML `div`, `body`, or `blockquote` element which was included in this document, so that that information (which is stored on a logical structure with no direct analog in WordprocessingML) can be maintained when an HTML document is stored in the WordprocessingML format.

The div element stores the following information about these structures:

- The child HTML `div`, and `blockquote` elements
- The borders for the element
- The margins for the element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

When the resulting WordprocessingML document is displayed by an application, the settings specified by this information shall be reflected in the formatting of the resulting paragraphs (i.e. this information shall not only be used when the document is resaved in the HTML format).

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style="border-left-style: solid; border-left-width: 1px; border-right-
style: solid; border-right-width: 1px; padding-left: 4px; padding-right: 4px;
padding-top: 1px; padding-bottom: 1px; margin-left: 50px">
      <p>Paragraph one.</p>
      <p>Paragraph two.</p>
    </div>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the `div`, `blockquote`, and body elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    <w:marLeft w:val="750" />
    …
    <w:divBdr>
      <w:left w:val="single" w:sz="6" w:color="auto" />
      <w:right w:val="single" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The div element specifies all margin and border information about the single HTML `div` structures in the document; in this case, the left indentation and the left and right borders. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

id (div Data ID)	Specifies a unique decimal number which shall be used to associate one or more structures in the WordprocessingML content with this HTML `div` information.
	
	When a WordprocessingML structure (a paragraph or a table row) is associated with `div` information, it shall be associated with the set of information which most immediately contains the current object.
	
	[*Example*: If a paragraph is wrapped within two HTML div elements, like this:
	
	`<div>`
	`  <div>`
	`  <p>Paragraph</p>`
	`  </div>`
	`</div>`
	
	The resulting WordprocessingML paragraph must reference the div Data ID associated with the inner HTML `div` element - the fact that it is also contained within the outer HTML `div` must be implied by the nesting of the corresponding WordprocessingML div elements in the web settings part. *end example*]
	
	The ID specified by this attribute is then referenced by the divId element for all structures which are immediately contained within the specified HTML `div`.
	
	[*Example*: Consider a simple HTML document defined as follows:
	
	`<html>`
	`  <body style=" margin-top:50px">`
	`  <p>Paragraph one.</p>`
	`  <div style="margin-left:50px">`
	`  <p>Paragraph two.</p>`
	`  </div>`
	`  </body>`
	`</html>`
	
	If the outer and inner body and `div` elements were assigned id attributes as follows:
	
	`<w:divs>`
	`  <w:div w:id="1626542603">`
	`  <w:bodyDiv w:val="1" />`
	`  …`
	`  <w:divsChild>`
	`  <w:div w:id="313534916">`
	`  …`
	`  </w:div>`
	`  </w:divsChild>`
	`  </w:div>`
	`</w:divs>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	Then the first paragraph would reference the div ID of the outer div (since it is contained by the HTML body element) and the second paragraph would reference the div ID of the inner div (since it is contained within the child HTML div element), as follows:
	
	`<w:p>`
	`  <w:pPr>`
	`  <w:divId w:val="1626542603" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Paragraph one.</w:t>`
	`  </w:r>`
	`</w:p>`
	`<w:p>`
	`  <w:pPr>`
	`  <w:divId w:val="313534916" />`
	`  </w:pPr>`
	`  <w:r>`
	`  <w:t>Paragraph one.</w:t>`
	`  </w:r>`
	`</w:p>`
	
	The id attributes on the div elements link each paragraph with the corresponding container div element. *end example*]
	
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Div) is located in §A.1. *end note*]

17.15.2.7 divBdr (Set of Borders for HTML div)

This element specifies the set of borders for the boundaries of the current HTML div, body, or blockquote element, using the four border types defined by its child elements.

If this element is omitted, then there shall be no borders associated with the current HTML v, body, or blockquote element.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style=" border-left-style:solid; border-right-style:groove; border-
right-width:3px; border-top-style:dashed; border-top-width:3px; border-bottom-
style:outset; border-bottom-width:3px">
      <p>paragraph of text</p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      </div>
    </body>
  </html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    …
    <w:divBdr>
      <w:top w:val="dashed" w:sz="18" w:space="7" w:color="auto" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="auto" />
      <w:bottom w:val="outset" w:sz="18" w:color="auto" />
      <w:right w:val="threeDEngrave" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The divBdr element specifies border information about the single HTML `div` structure in the document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DivBdr) is located in §A.1. *end note*]

17.15.2.8 divs (Information about HTML div Elements)

This element specifies all information about the set of HTML `div` elements (as well as the `body` and `blockquote` elements) which were included in this document, so that that information (which is stored on a logical structure with no direct analog in WordprocessingML) can be maintained when an HTML document is stored in the WordprocessingML format.

The divs element stores the following information about these structures:

- The parent/child structure of HTML `div`, `blockquote`, and `body` elements
- The borders for each of these elements
- The margins for each of these elements

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

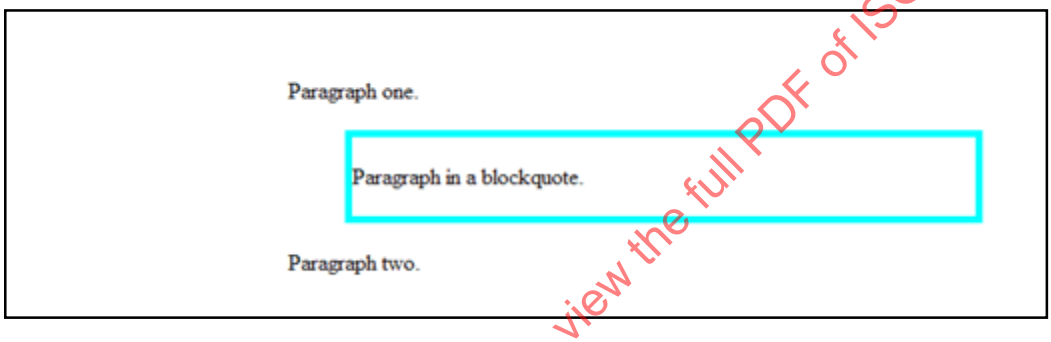
 ©ISO/IEC 2016 – All rights reserved

When the resulting WordprocessingML document is displayed by an application, the settings specified by this information shall be reflected in the formatting of the resulting paragraphs (i.e. this information shall not only be used when the document is resaved in the HTML format).

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body style="margin-left:200px;margin-top:50px">
    <p>Paragraph one.</p>
    <blockquote style="border: 5px solid #00FFFF">
      <p>Paragraph in a blockquote.</p>
    </blockquote>
    <p>Paragraph two.</p>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):



Now, when this document is saved in the WordprocessingML format, the information stored on the `div`, `blockquote`, and body elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1626542603">
    <w:bodyDiv w:val="1" />
    <w:marLeft w:val="3000" />
    <w:marTop w:val="750" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
…    <w:divsChild>
       <w:div w:id="313534916">
         <w:blockQuote w:val="1" />
         <w:marLeft w:val="720" />
         <w:marRight w:val="720" />
         <w:marTop w:val="100" />
         <w:marBottom w:val="100" />
         <w:divBdr>
           <w:top w:val="single" w:sz="36" w:color="00FFFF" />
           <w:left w:val="single" w:sz="36" w:color="00FFFF" />
           <w:right w:val="single" w:sz="36" w:color="00FFFF" />
           <w:bottom w:val="single" w:sz="36" w:color="00FFFF" />
         </w:divBdr>
       </w:div>
     </w:divsChild>
   </w:div>
 </w:divs>
```

The divs element specifies all of the margin and border information about the necessary HTML structures in the document; in this case, the body element and the nested blockquote. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Divs) is located in §A.1. *end note*]

17.15.2.9 divsChild (Child div Elements Contained within Current div)

This element specifies the set of HTML div or blockquote elements which are contained within the current HTML div, body, or blockquote element, establishing the parent/child hierarchy of the original set of these elements.

When an HTML document containing these objects is saved in the WordprocessingML format, WordprocessingML objects store a reference to their most immediate parent div, body, or blockquote element using the divId element.

However, since only a single reference is stored, this information is often insufficient to determine the appropriate parent/child hierarchy for the original HTML div data, so it can be applied appropriately. This element allows that hierarchy to be stored, as child HTML div elements are stored within the childDivs element.

[*Example*: Consider a simple HTML document defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<html>
  <body>
    <div style=" margin-top:50px">
      <p>Paragraph one.</p>
      <div style="margin-left:50px">
        <p>Paragraph two.</p>
      </div>
    </div>
  </body>
</html>
```

If the outer and inner body and div elements were assigned id attributes of 1626542603 and 313534916 respectively, then the first paragraph would reference the div ID of the outer div (since it is contained within that HTML div element) and the second paragraph would reference the div ID of the inner div (since it is contained within the child HTML div element), as follows:

```
<w:p>
  <w:pPr>
    <w:divId w:val="1626542603" />
  </w:pPr>
  <w:r>
    <w:t>Paragraph one.</w:t>
  </w:r>
</w:p>
<w:p>
  <w:pPr>
    <w:divId w:val="313534916" />
  </w:pPr>
  <w:r>
    <w:t>Paragraph one.</w:t>
  </w:r>
</w:p>
```

However, this information alone is insufficient - it is unclear if the second div is contained within, or simply adjacent to, the first one.

In order to preserve this information, the correct hierarchy is stored within the web settings part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:divs>
  <w:div w:id="1626542603">
    …
    <w:divsChild>
      <w:div w:id="313534916">
        …
      </w:div>
    </w:divsChild>
  </w:div>
</w:divs>
```

The divsChild element contains the second div as a child of the first div, specifying that the first div covers both paragraphs. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Divs) is located in §A.1. *end note*]

17.15.2.10 doNotOrganizeInFolder (Do Not Place Supporting Files in Subdirectory)

This element specifies that applications shall not automatically place all supporting files (images which are part of this HTML web page, etc.) in a subdirectory when the contents of this WordprocessingML document are saved as a web page. Typically, applications which save a document as a web page consisting of multiple files save all supporting files in a subdirectory next to the main HTML file (in order to keep those files organized). This element specifies the files shall be placed in the same directory as the actual web page.

If this element is omitted from the document, then all supporting files should be saved into a subdirectory beneath the main web page file when this document is saved as a web page.

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:doNotOrganizeInFolder w:val="true" />
</w:webSettings>
```

The doNotOrganizeInFolder element has a val attribute value of `true`, which specifies that applications should save all supplementary files in the same directory as the main web page HTML document when saving this WordprocessingML document as a web page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.11 doNotRelyOnCSS (Do Not Rely on CSS for Font Face Formatting)

This element specifies whether applications can rely on the CSS properties for font face (the `font-family` property) when saving this WordprocessingML document as a web page. If this element is utilized, then the HTML `font` element should be used either in place of or in concert with these CSS properties in order to specify the font face formatting for the resulting web page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then applications can choose to rely on the CSS properties for font face as desired.

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of these CSS properties when attempting to read and display the resulting web page, in order to maximize the fidelity of the resulting output. *end note*]

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:doNotRelyOnCSS w:val="true" />
</w:webSettings>
```

The doNotRelyOnCSS element has a val attribute value of `true`, which specifies that applications should include the HTML `font` element when saving this WordprocessingML document as a web page. For example, this output:

```
<span style='font-family:"Courier New"'>text</span>
```

This output would instead be saved as follows:

```
<font face="Courier New"><span style='font-family:"Courier
New"'>text</span></font>
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.12 doNotSaveAsSingleFile (Recommend Web Page Format over Single File Web Page Format)

This element specifies that applications should recommend that new web page files generated using this WordprocessingML document use a multi-file web page format (HTML), rather than a single-file web page format (MHTML) when this document is saved as an HTML web page. This setting shall not prevent the use of the MHTML format; it shall only cause applications to recommend (via a default) a non single-file format when saving as a web page.

[*Note*: This setting is primarily intended for applications which explicitly support a "Save as Web Page…" action, in order to determine the default setting for the resulting web page. *end note*]

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:doNotSaveAsSingleFile w:val="true" />
</w:webSettings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The doNotSaveAsSingleFile element specifies that applications should recommend a multi-file web page format when this document is subsequently saved as a web page. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.13 doNotUseLongFileNames (Do Not Use File Names Longer than 8.3 Characters)

This element specifies that applications shall ensure that the file names for all files generated when saving this document as a web page do not exceed eight octets with a three octet extension. This includes all supporting files (images which are part of this HTML web page, etc.). The file names generated are not case-sensitive.

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of long file names when attempting to read and display the resulting web page. *end note*]

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:doNotUseLongFileNames w:val="true" />
</w:webSettings>
```

The doNotUseLongFileNames element specifies that applications should ensure that all file names generated when this document is subsequently saved as a web page do not exceed the 8.3 octet file name limitation. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.14 encoding (Output Encoding When Saving as Web Page)

This element specifies the encoding which shall be used for the contents of this WordprocessingML document when it is saved as an HTML web page. The set of encodings supported by this element shall be derived from the standard set of character set definitions provided at http://www.iana.org/assignments/character-sets.

If this element is omitted, then the default encoding for the current system shall be used when this document is saved as a web page. If the value of the val attribute is unknown or supported by an application, then the default encoding for the current system shall be used when this document is saved as a web page.

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:encoding w:val="utf-8" />
</w:webSettings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The encoding element's val attribute has a value of `utf-8`, which specifies that this document must be encoded in the UTF-8 format when it is saved as a web page. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.2.15 flatBorders (Frameset Splitter Border Style)

This element specifies the 3D style of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and can be ignored for all nested framesets in this document. When this property is turned on, the borders for this frameset shall be flat (not 3D), otherwise they can be presented as 3D splitter when they are displayed.

If this element is omitted, then the default style of the splitter should be a 3D splitter.

[*Example*: Consider a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The following properties define the presentation of the splitter bars within this frameset:

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="200" />
    <w:color w:val="0000FF" />
    <w:flatBorders w:val="true" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

The flatBorders element's val attribute has a value of `true`, which specifies that the style of the splitters must be flat (the splitter might not be 3D when displayed). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

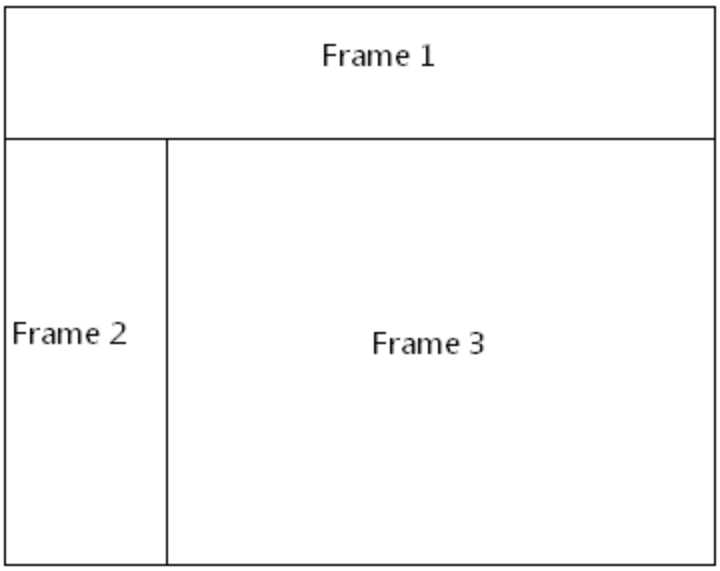
17.15.2.16 frame (Single Frame Properties)

This element specifies the properties for a single frame within a frameset document. When a document defines a frameset using the frameset element; that frameset is composed of a set of frames, each of which is specified by a single frame element.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frame>
    <w:sz w:val="20%" />
    <w:name w:val="Frame 1" />
    <w:sourceFileName r:id="rId1" />
  </w:frame>
  <w:frameset>
    …
  </w:frameset>
</w:frameset>
```

The frame element specifies the set of properties for a single frame in the document. In this case, these properties (for the frame marked with `Frame 1` in the diagram above) specify that the frame must have the following properties:

- A height of `20%` of the height of the document
- A name of `Frame 1`
- The contents of the frame must be pulled from the document that is the target of the relationship with ID `rId1`

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Frame) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.2.17 frameLayout (Frameset Layout)

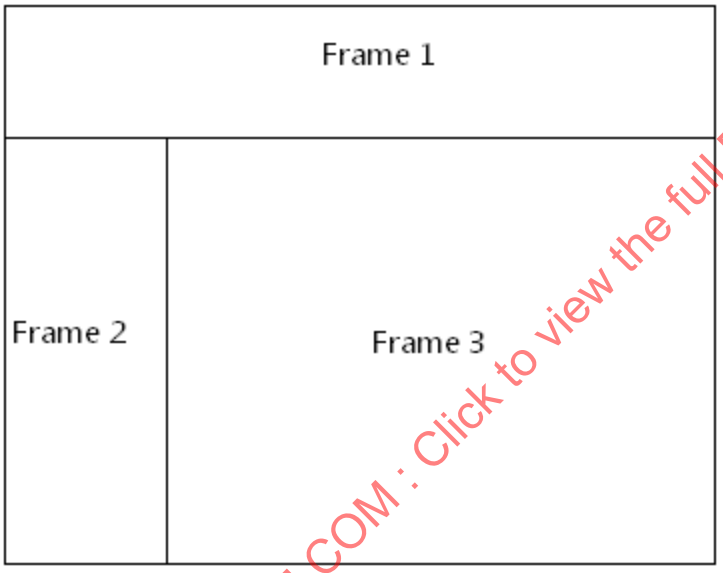
This element specifies the order in which the frames (and nested framesets) in a frameset shall be displayed. When a frameset is created, it can only contain frames which are stacked in one direction:

- Vertically (one on top of another)
- Horizontally (one next to another)

This element specifies how the frames in this frameset are stacked, which shall also be used to interpret the sizes defined by the sz element (§17.15.2.39) for each frame. In order to determine the ordering of the constituent frames within this frameset, the ordering of the child frame and frameset elements shall be used.

If this element is omitted, then the frames in this frameset shall be stacked vertically on top of one another (a row frameset).

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frameLayout w:val="rows" />
  <w:frame>
    …
  </w:frame>
  <w:frameset>
    <w:frameLayout w:val="cols" />
    <w:frame>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      …
    </w:frame>
    <w:frame>
      …
    </w:frame>
  </w:frameset>
</w:frameset>
```

The frameLayout element specifies that the outer frameset is a consists of the single frame and the child frameset stacked vertically, and an inner nested frameset consisting of two frames stacked horizontally. *end example*]

Attributes	Description

val (Frameset Layout Value)	Specifies the type of layout which shall be used to display the contents of the frames and nested framesets within this frameset, as defined by the simple type referenced below.
	
	[*Example*: Consider a frameset definition within a WordprocessingML document which defines the following frameset layout setting:
	
	`<w:frameset>`
	`  <w:frameLayout w:val="cols" />`
	`  …`
	`</w:frameset>`
	
	The val attribute value of `cols` specifies that the contents of this frameset must be stacked horizontally (in columns). *end example*]
	
	The possible values for this attribute are defined by the ST_FrameLayout simple type (§17.18.31).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FrameLayout) is located in §A.1. *end note*]

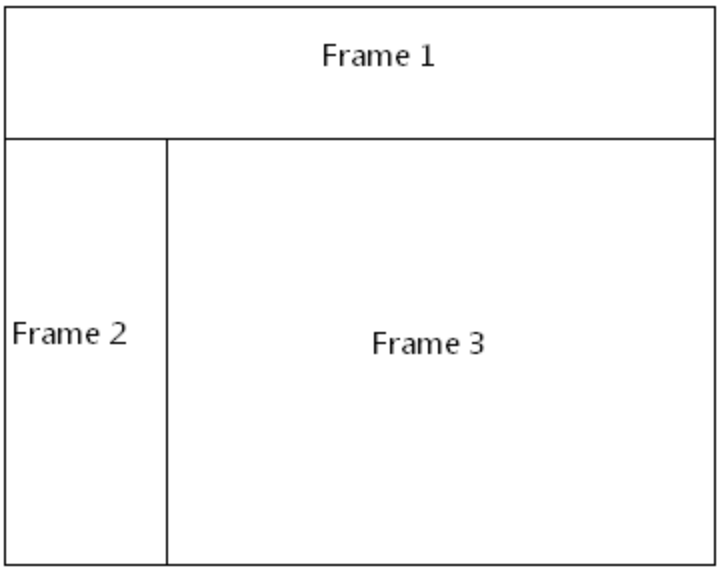
17.15.2.18 frameset (Nested Frameset Definition)

This element specifies a frameset which has been nested within another frameset within a WordprocessingML document. This WordprocessingML element is analogous to the `frameset` element in HTML (when that frameset is the child of another `frameset` element).

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  <w:frameLayout w:val="rows" />
  <w:frame>
    …
  </w:frame>
  <w:frameset>
    <w:frameLayout w:val="cols" />
    <w:frame>
      …
    </w:frame>
    <w:frame>
      …
    </w:frame>
  </w:frameset>
</w:frameset>
```

The child frameset element specifies the frameset definition for the inner frameset; that frameset consists of two frames (`Frame 2` and `Frame 3` in the image above) which have been laid out horizontally as columns. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Frameset) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

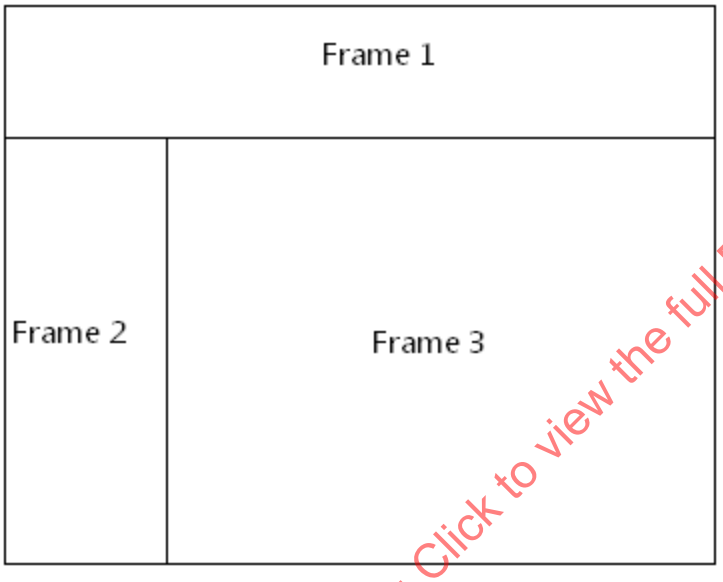
17.15.2.19 frameset (Root Frameset Definition)

This element specifies that this document is the container for a frameset. This WordprocessingML element is analogous to the `frameset` element in HTML.

When the frameset element is present within a document, that document shall serve as a frameset definition only; all of its normal document content shall therefore not be displayed as long as it contains at least one child frame or frameset element.

If this element is omitted, then the currently document shall not be treated as a frameset definition; its regular document content shall be displayed.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  <w:frameLayout w:val="rows" />
  <w:frame>
    …
  </w:frame>
  <w:frameset>
    <w:frameLayout w:val="cols" />
    <w:frame>
      …
    </w:frame>
    <w:frame>
      …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      </w:frame>
    </w:frameset>
  </w:frameset>
```

The parent frameset element specifies that the current document is a frameset definition; that frameset consists of a single frame and another nested frameset stacked vertically. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Frameset) is located in §A.1. *end note*]

17.15.2.20 framesetSplitbar (Frameset Splitter Properties)

This element specifies the properties for the splitters associated with this frameset. A *splitter* is a horizontal or vertical line which visually separates the contents of one frame from another within a frameset.

If this element is omitted, then the default parameters for each of the child frameset properties shall be used for all splitters in this frameset.

[*Example*: Consider a frameset consisting of the following three frames:

The following properties define the presentation of the splitter bars within this frameset:

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="90" />
    <w:color w:val="auto" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The framesetSplitbar element specifies the properties for all splitters in this frameset; in this case, those properties are that the splitter must be 4.5 points (90 twentieths of a point) wide, and that the color of the splitter must be automatically determined via the attribute value of auto. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FramesetSplitbar) is located in §A.1. *end note*]

17.15.2.21 left (Left Border for HTML div)

This element specifies the border which shall be displayed at the left of the boundaries of the current HTML div object.

If this element is omitted, then this HTML div object shall not have a left border.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style=" border-left-style:solid; border-right-style:groove; border-
right-width:1px; border-top-style:dashed; border-top-width:3px; border-bottom-
style:outset; border-bottom-width:3px">
      <p>paragraph of text</p>
    </div>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the div elements is stored in the web setting part as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:divs>
  <w:div w:id="1785730240">
     …
    <w:divBdr>
      <w:top w:val="dashed" w:sz="18" w:space="7" w:color="auto" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="auto" />
      <w:bottom w:val="outset" w:sz="18" w:color="auto" />
      <w:right w:val="threeDEngrave" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The left element specifies border information about the left border for the single HTML div structure in the document; in this case, a 3 point bottom border of type `single`. The initial 4 pixel border was converted to 3 points using the following logic:

$$4\text{px} * \frac{1 \text{ inch}}{96 \text{ px}} * \frac{576 \text{ eighth points}}{1 \text{ inch}} = 24 \text{ eighth points (3 points)}$$

end example]

This element's content model is defined by the common border properties definition in §17.3.4.

17.15.2.22 linkedToFile (Maintain Link to Existing File)

This element specifies that the file referenced by the sourceFileName element (§17.15.2.38) as the basis for the current frame shall not be changed, even when the file defined by the parent frameset is moved - i.e. the link shall remain exactly as specified.

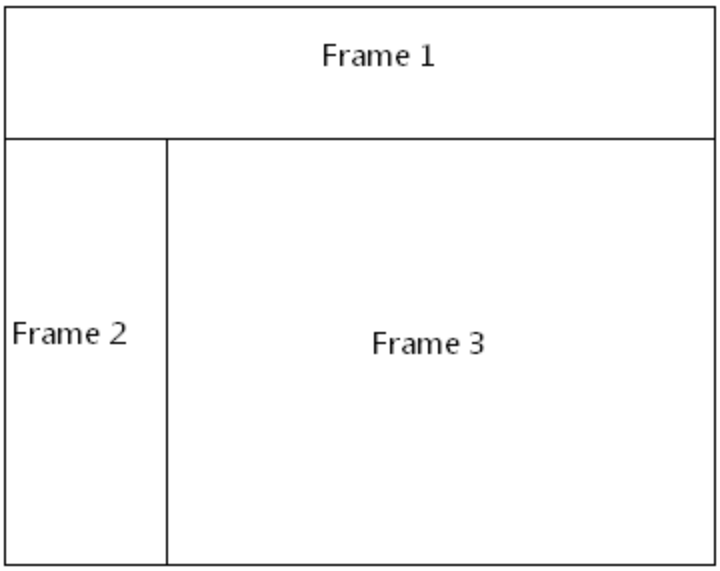
[*Guidance*: Typically, when a document is incorporated into a frameset, a copy of that document is made such that all files encompassing the frameset are stored in a single subdirectory (so they can be moved as a single unit). However, if the link to the current file is absolute and must not be changed even when the location of the main frameset document, then this element must be set to indicate that setting. *end guidance*]

If this element is omitted, then a new file can be created as necessary when the parent frameset document is resaved to another location.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frame>
    <w:sz w:val="20%" />
    <w:name w:val="Frame 1" />
    <w:sourceFileName r:id="rId1" />
    <w:linkedToFile w:val="true" />
  </w:frame>
  <w:frameset>
    …
  </w:frameset>
</w:frameset>
```

The linkedToFile element specifies that the frame source location specified by the sourceFileName element (§17.15.2.38) must not be modified, even if the parent frameset document is resaved to another location. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

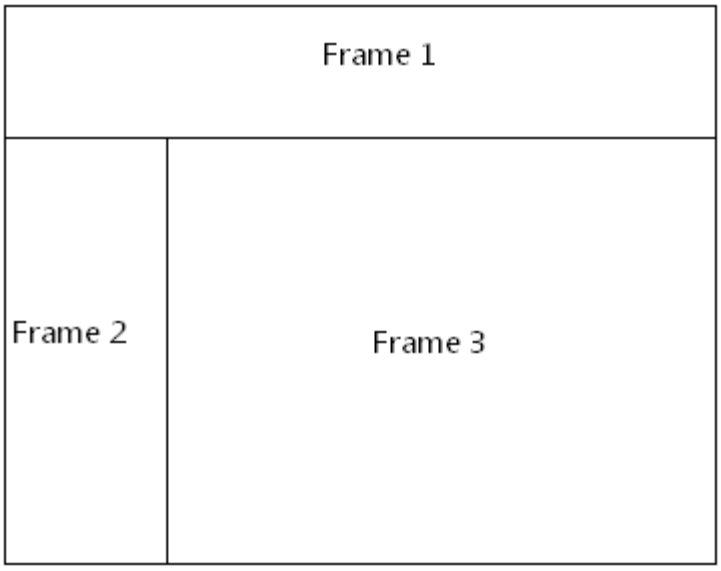
17.15.2.23 longDesc (Frame Long Description)

This element specifies an explicit relationship whose target is the long description of the frame. This description should supplement the short description provided by the title element. This property is analogous to the longdesc attribute on the frame element in HTML.

If this element is omitted, then no long description shall be associated with the given frame.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frame>
    <w:name w:val="Frame 1" />
    <w:title w:val="Menu bar" />
    <w:longDesc r:id="rIdMenuBar" />
  </w:frame>
  <w:frameset>
    <w:title w:val="Navigation and document collection" />
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:title w:val="Navigation bar" />
      <w:longDesc r:id="rIdNavBar" />
    </w:frame>
    <w:frame>
      <w:name w:val="Frame 3" />
      <w:title w:val="Documents" />
      <w:longDesc r:id="rIdDocs" />
    </w:frame>
  </w:frameset>
</w:frameset>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The longDesc element specifies that the part targeted by the relationship with an id of rIdMenuBar must be used for supplementary information for Frame 1. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="rIdMenuBar" TargetMode="External"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink"
Target="myexample-desc.html#menubar" />
  …
</Relationships>
```

The corresponding relationship part item shows that the long description supplementary information for Frame 1 is located at myexample-desc.html#menubar, where myexample-desc.html describes #menubar as "This frame provides links to the major sections of the site: Home, Resources, Links, Help." *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.2.24 marBottom (Bottom Margin for HTML div)

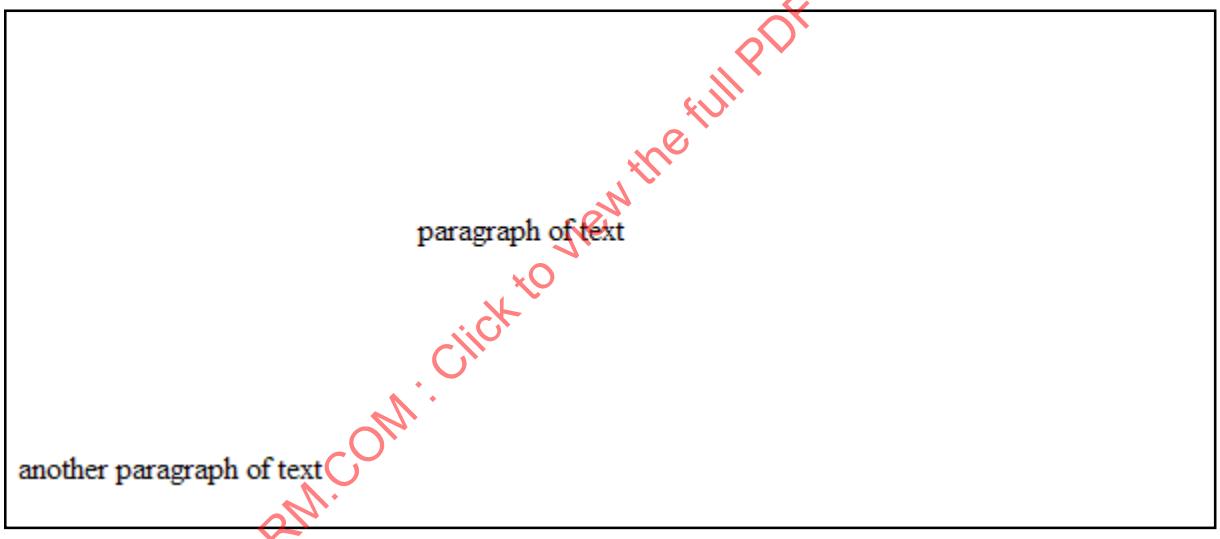
This element specifies the margin which shall be displayed at the bottom of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a bottom margin.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style="margin-top:100px; margin-left:200px; margin-right:50px; margin-
bottom:100px">
      <p>paragraph of text</p>
    </div>
    <p>another paragraph of text</p>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):



Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    <w:marLeft w:val="3000" />
    <w:marRight w:val="750" />
    <w:marTop w:val="1500" />
    <w:marBottom w:val="1500" />
  </w:div>
</w:divs>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The marBottom element specifies margin information about the bottom margin for the single HTML `div` structure in the document; in this case, a 75 point bottom margin. The initial 100 pixel margin was converted to 75 points using the following logic:

$$100\text{px} * \frac{1 \text{ inch}}{96 \text{ px}} * \frac{1440 \text{ twentieth points}}{1 \text{ inch}} = 1500 \text{ twentieth points (75 points)}$$

end example]

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of `-720` whose type is ST_SignedTwipsMeasure. This attribute value specifies a value of negative one-half of an inch or -36 points (`-720` twentieths of a point = -36 points = -0.5 inches). *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

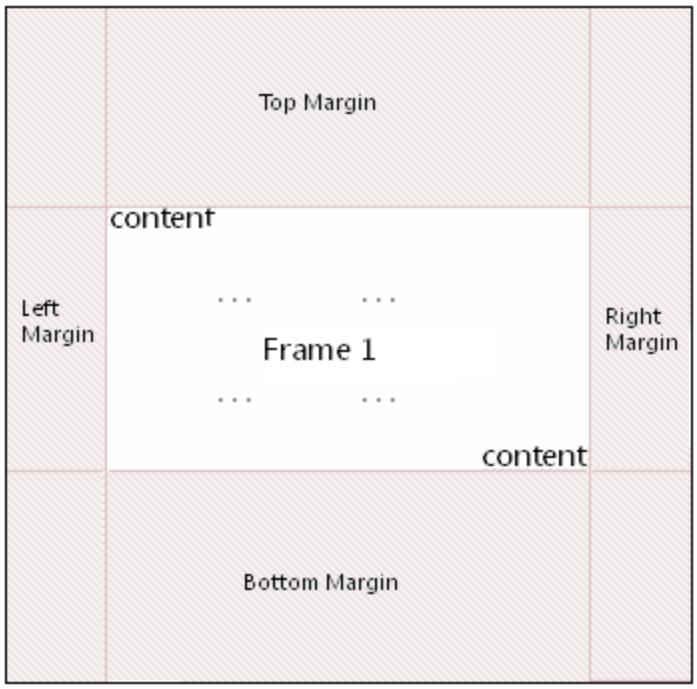
[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedTwipsMeasure) is located in §A.1. *end note*]

17.15.2.25 marH (Top and Bottom Margin for Frame)

This element specifies the top and bottom margin height for a single frame in a frameset document, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



This height is expressed in pixels.

If this element is omitted, then no top or bottom margin shall be used for this frame.

[*Example*: Consider a document that has a frame, where the margin height has been specified and is represented as the following WordprocessingML:

```
<w:frame>
  <w:marH w:val="594"/>
</w:frame>
```

The marH element has a val attribute value of 594, which specifies that this frame has a top and bottom margin value of 594 pixels, resulting in 594 pixels of space between the content and the top and bottom margins of the frame. *end example*]

Attributes	Description
val (Measurement in Pixels)	Specifies a value whose contents shall contain a positive whole number, whose contents consist of a positive measurement in pixels. The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of 960 whose simple type is ST_PixelsMeasure. This attribute value specifies a value of 960 pixels. *end example*] The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§17.18.67).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_PixelsMeasure) is located in §A.1. *end note*]

17.15.2.26 marLeft (Left Margin for HTML div)

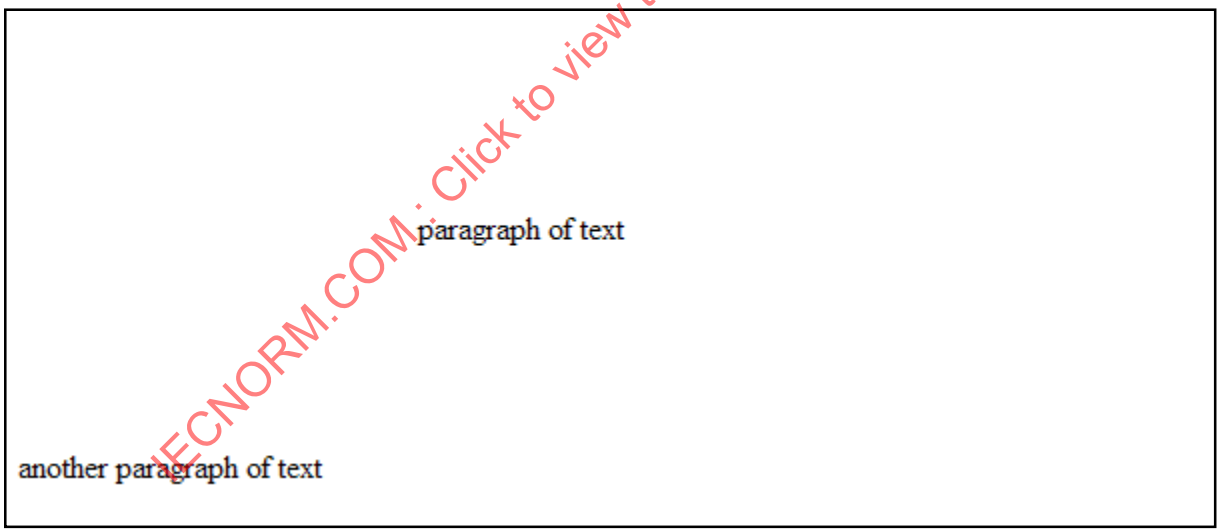
This element specifies the margin which shall be displayed at the left of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a left margin.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style="margin-top:100px; margin-left:200px; margin-right:50px; margin-
bottom:100px">
      <p>paragraph of text</p>
    </div>
    <p>another paragraph of text</p>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):



Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:divs>
  <w:div w:id="1785730240">
    <w:marLeft w:val="3000" />
    <w:marRight w:val="750" />
    <w:marTop w:val="1500" />
    <w:marBottom w:val="1500" />
  </w:div>
</w:divs>
```

The marLeft element specifies margin information about the left margin for the single HTML `div` structure in the document; in this case, a 150 point left margin. The initial 200 pixel margin was converted to 150 points using the following logic:

$$200\text{px} * \frac{1 \text{ inch}}{96 \text{ px}} * \frac{1440 \text{ twentieth points}}{1 \text{ inch}} = 3000 \text{ twentieth points } (150 \text{ points})$$

end example]

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of `-720` whose type is ST_SignedTwipsMeasure. This attribute value specifies a value of negative one-half of an inch or -36 points (`-720` twentieths of a point = -36 points = -0.5 inches). *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedTwipsMeasure) is located in §A.1. *end note*]

17.15.2.27 marRight (Right Margin for HTML div)

This element specifies the margin which shall be displayed at the right of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a right margin.

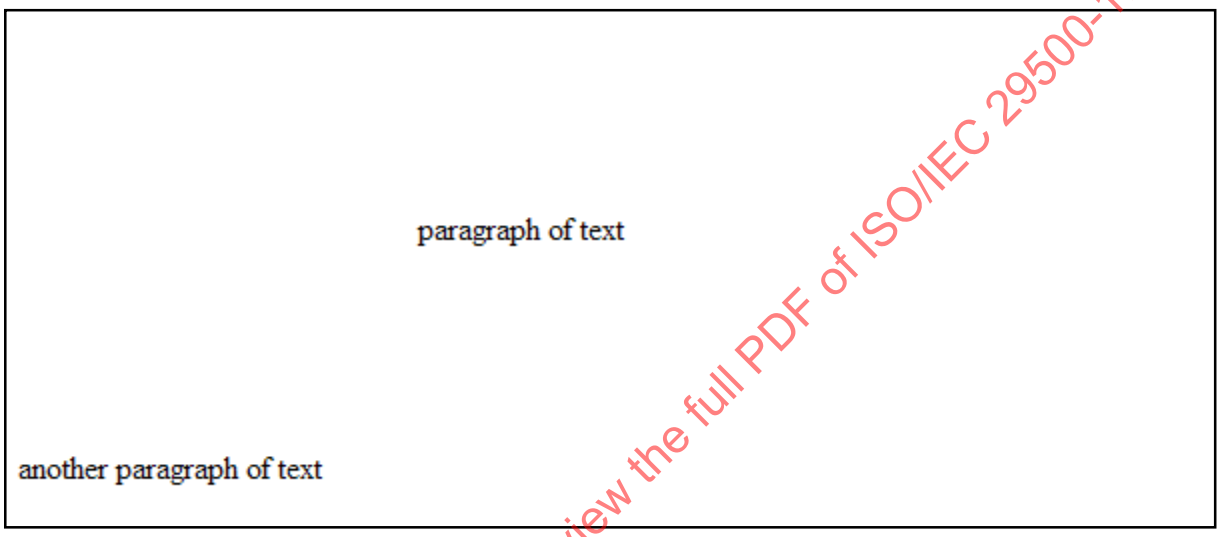
[*Example*: Consider a simple HTML document defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<html>
  <body>
    <div style="margin-top:100px; margin-left:200px; margin-right:50px; margin-
bottom:100px">
      <p>paragraph of text</p>
    </div>
    <p>another paragraph of text</p>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):



Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    <w:marLeft w:val="3000" />
    <w:marRight w:val="750" />
    <w:marTop w:val="1500" />
    <w:marBottom w:val="1500" />
  </w:div>
</w:divs>
```

The marRight element specifies margin information about the right margin for the single HTML `div` structure in the document; in this case, a 37.5 point right margin. The initial 50 pixel margin was converted to 37.5 points using the following logic:

$$50\text{px} * \frac{1 \text{ inch}}{96 \text{ px}} * \frac{1440 \text{ twentieth points}}{1 \text{ inch}} = 750 \text{ twentieth points (37.5 points)}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of `-720` whose type is ST_SignedTwipsMeasure. This attribute value specifies a value of negative one-half of an inch or -36 points (`-720` twentieths of a point = -36 points = -0.5 inches). *end example*] The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedTwipsMeasure) is located in §A.1. *end note*]

17.15.2.28 marTop (Top Margin for HTML div)

This element specifies the margin which shall be displayed at the top of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a top margin.

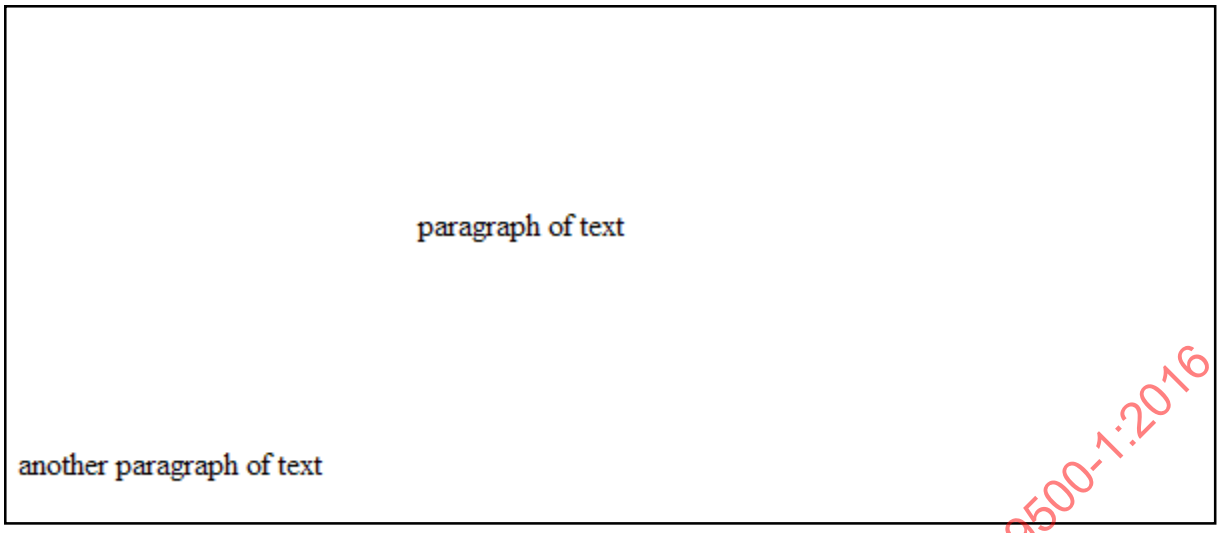
[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style="margin-top:100px; margin-left:200px; margin-right:50px; margin-
bottom:100px">
      <p>paragraph of text</p>
    </div>
    <p>another paragraph of text</p>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    <w:marLeft w:val="3000" />
    <w:marRight w:val="750" />
    <w:marTop w:val="1500" />
    <w:marBottom w:val="1500" />
  </w:div>
</w:divs>
```

The marTop element specifies margin information about the top margin for the single HTML `div` structure in the document; in this case, a 75 point top margin. The initial 100 pixel margin was converted to 75 points using the following logic:

$$100\text{px} * \frac{1\text{ inch}}{96\text{ px}} * \frac{1440\text{ twentieth points}}{1\text{ inch}} = 1500\text{ twentieth points (75 points)}$$

end example]

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of `-720` whose type is ST_SignedTwipsMeasure. This attribute value specifies a value of negative one-half of an inch or -36 points (`-720` twentieths of a point = -36 points = -0.5 inches). *end example*]

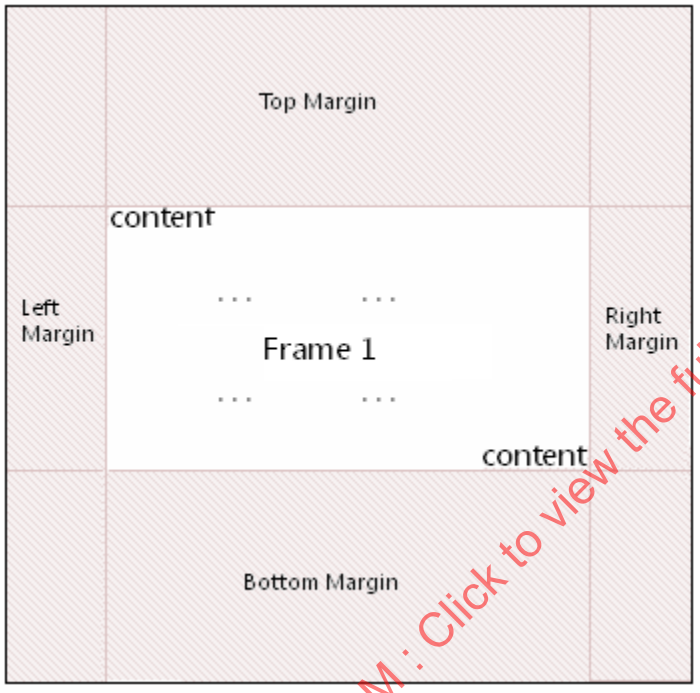
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§17.18.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SignedTwipsMeasure) is located in §A.1. *end note*]

17.15.2.29 marW (Left and Right Margin for Frame)

This element specifies the left and right margin height for a single frame in a frameset document, as follows:



This height is expressed in pixels.

If this element is omitted, then no left or right margin shall be used for this frame.

[*Example*: Consider a document that has a frame, where the frame's margins have been specified and is represented as the following WordprocessingML:

```
<w:frame>
  <w:marW w:val="294"/>
</w:frame>
```

The marW element has a val attribute value of 294, which specifies that this frame has a left and right margin value of 294 pixels, resulting in 294 pixels of space between the content and the left and right margins of the frame. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Measurement in Pixels)	Specifies a value whose contents shall contain a positive whole number, whose contents consist of a positive measurement in pixels. The contents of this measurement shall be interpreted based on the context of the parent XML element. [*Example*: Consider an attribute value of `960` whose simple type is ST_PixelsMeasure. This attribute value specifies a value of 960 pixels. *end example*] The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§17.18.67).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PixelsMeasure) is located in §A.1. *end note*]

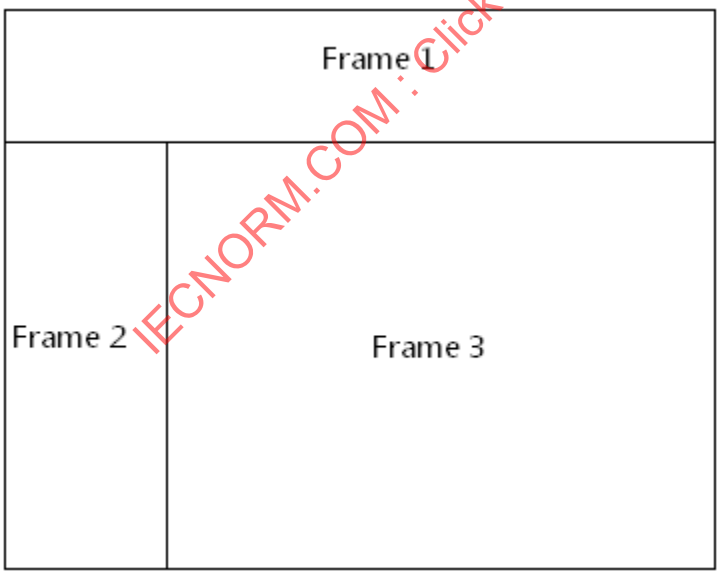
17.15.2.30 name (Frame Name)

This element specifies the name of a single frame within a frameset document. This property is analogous to the `name` attribute on the `frame` element in HTML.

[*Note*: The name of a frame can be used in web pages that reference a frame via targeted links, etc. *end note*]

If this element is omitted, then the current frame shall have no name associated with it.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frame>
    <w:name w:val="Frame 1" />
  </w:frame>
  <w:frameset>
    …
    <w:frame>
      <w:name w:val="Frame 2" />
    </w:frame>
    <w:frame>
      <w:name w:val="Frame 3" />
    </w:frame>
  </w:frameset>
</w:frameset>
```

The name element specifies the name for each frame within this frameset; in this case, the frames have names of `Frame 1`, `Frame 2`, and `Frame 3` respectively. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.2.31 noBorder (Do Not Display Frameset Splitters)

This element specifies whether the splitters shall be displayed for the contents of the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and can be ignored for all nested framesets in this document. If this element is present, then no splitters shall be displayed, and all other frameset splitter properties can be ignored.

If this element is omitted, then the splitters in this document shall be displayed as defined by the w and color elements.

[*Example*: Consider a frameset consisting of the following three frames:

The following properties define the presentation of the splitter bars within this frameset:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="200" />
    <w:color w:val="0000FF" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

If the noBorder element is also specified:

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="200" />
    <w:color w:val="0000FF" />
    <w:noBorder w:val="true" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

Then all frameset splitters are suppressed:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The noBorder element's val attribute has a value of `true`, which specifies that the splitters for this document must not be displayed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

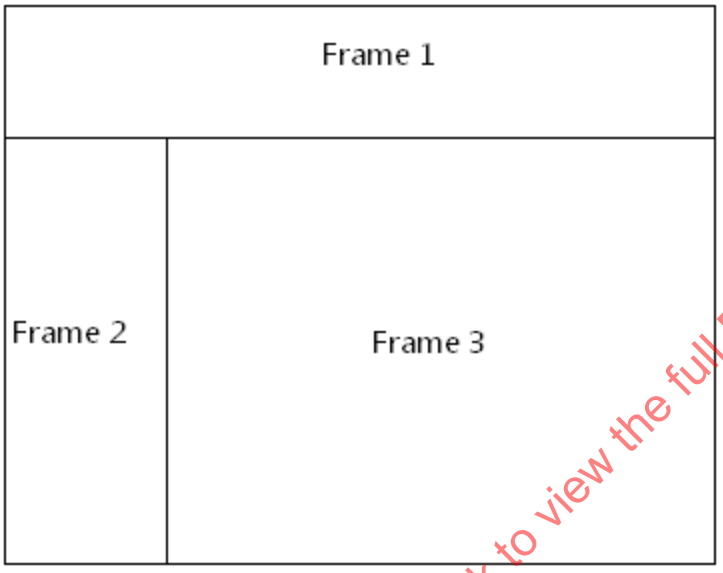
©ISO/IEC 2016 – All rights reserved

17.15.2.32 noResizeAllowed (Frame Cannot Be Resized)

This element specifies whether or not the size of the current frame shall be modifiable (i.e. whether the frame can be resized) when the contents of this document are saved as HTML and displayed in a web browser. When this element is set, the size of the frame shall be set to its current values. This property is analogous to the `noresize` attribute on the `frame` element in HTML.

If this element is omitted, the size of the frame shall be modifiable (the frame can be resized when it is displayed).

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frameset>
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:noResizeAllowed w:val="true" />
    </w:frame>
    …
  </w:frameset>
</w:frameset>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The noResizeAllowed element has a val attribute of `true`, which specifies that the size of the frame specified by `Frame 2` must not be modifiable (the two borders which intersect that frame cannot be resized). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.33 optimizeForBrowser (Disable Features Not Supported by Target Web Profile)

This element specifies whether applications should attempt to customize the output for any web page produced from this document, as well as the HTML output to which it should be customized. [*Example*: This might involve blocking any output which is not supported by that target output profile. *end example*] The target output profile is identified by the contents of the target attribute.

[*Example*: Consider a document whose web settings part contains the following WordprocessingML:

```
<w:webSettings>
    …
    <w:optimizeForBrowser w:target="W3C XHTML+CSS1" />
    <w:allowPNG w:val="on"/>
</w:webSettings>
```

The optimizeForBrowser element specifies that the output should be optimized for a specific output target. Since the value of the target attribute specifies a target of W3C XHTML 1.0 output using CSS 1, any HTML generated should fall within that profile. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>target (Target Output Profile)</td><td>Specifies the version of HTML output targeted by the output of any web page produced by this document. This attribute shall only contain a string that represents an output profile defined by published standards and W3C recommendations. Product names shall not be used to define a profile.

The following reserved values and their targets are listed below:

<table>
<tr><th>Value</th><th>Target</th></tr>
<tr><td>W3C XHTML+CSS1</td><td>W3C XHTML 1.0 + CSS 1</td></tr>
<tr><td>W3C HTML4+CSS1</td><td>W3C HTML 4.01 + CSS 1</td></tr>
<tr><td>W3C XHTML+CSS2</td><td>W3C XHTML 1.0 + CSS 2</td></tr>
<tr><td>W3C HTML4+CSS2</td><td>W3C HTML 4.01 + CSS 2</td></tr>
</table>

If this attribute is omitted, the version of HTML output is application-defined.

[Example: For example, consider the following web publishing settings:

<code><w:optimizeForBrowser w:target="W3C XHTML+CSS2"/></code></td></tr>
</table>

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The target attribute explicitly declares that any web page generated from this document should target W3C XHTML 1.0 + CSS 2. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
val (On/Off Value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property shall be explicitly turned off. [*Example*: For example, consider the following on/off property: `<… w:val="false"/>` The val attribute explicitly declares that the property is `false`. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OptimizeForBrowser) is located in §A.1. *end note*]

17.15.2.34 pixelsPerInch (Pixels per Inch for Graphics/Images)

This element specifies the number of pixels per inch (or density) that is used for the display of pictures or table cells when a WordprocessingML document is saved as a web page. The size that is specified by this element affects the size of the pictures or table cells relative to the size of text in the document. The pixels per inch (ppi) measurement is relative to the screen resolution, and the resulting physical dimensions of the resulting image or cell in pixels (which are used in web pages, but not for printed documents) are the result of the original dimensions (in inches) multiplied by the number of pixels per inch.

The range of values for this element is typically from 19 to 480 pixels per inch. The common settings for popular screen sizes are 72, 96, and 120 pixels per inch.

If this element is omitted, then a default size of 96 pixels per inch shall be used when determining the number of pixels for images and/or table cells within this document.

[*Note*: This setting is typically only specified if the target screen resolution for the web page is known, as defined by the targetScreenSz element (§17.15.2.41) to set the optimum screen size for the web page. *end note*]

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:webSettings>
  <w:pixelsPerInch w:val="200" />
</w:webSettings>
```

The pixelsPerInch element's val attribute has a value of 200, which specifies that all inches to pixels conversions done when saving this web page must be done assuming a transformation of 200 pixels per inch. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: <… w:val="1512645511" /> The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.15.2.35 right (Right Border for HTML div)

This element specifies the border which shall be displayed at the right of the boundaries of the current HTML div object.

If this element is omitted, then this HTML div object shall not have a right border.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style=" border-left-style:solid; border-right-style:groove; border-
right-width:1px; border-top-style:dashed; border-top-width:3px; border-bottom-
style:outset; border-bottom-width:3px">
      <p>paragraph of text</p>
    </div>
  </body>
</html>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    …
    <w:divBdr>
      <w:top w:val="dashed" w:sz="18" w:space="7" w:color="auto" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="auto" />
      <w:bottom w:val="outset" w:sz="18" w:color="auto" />
      <w:right w:val="threeDEngrave" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The right element specifies border information about the right border for the single HTML `div` structure in the document; in this case, a 0.75 point bottom border of type `threeDEngrave`. The initial 1 pixel border was converted to 0.75 points using the following logic:

$$1\text{px} * \frac{1\text{ inch}}{96\text{ px}} * \frac{576\text{ eighth points}}{1\text{ inch}} = 6\text{ eighth points } (0.75\text{ points})$$

end example]

This element's content model is defined by the common border properties definition in §17.3.4.

17.15.2.36 saveSmartTagsAsXml (Save Smart Tag Data in XML Property Bag)

This element specifies that the information pertaining to all smart tags () in the current document shall be saved into a separate XML-based property bag at the head of the web page when this WordprocessingML document is saved as a web page.

[*Rationale*: This setting is typically used when saving documents known to contain smart tags as web pages, in order to allow the smart tag data within the web page to be processed as a separate XML document by a separate parser, even though the actual HTML content of the resulting web page cannot be parsed by an XML-based parser. *end rationale*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, then the smart tag data of this document shall not be saved into a separate XML-compliant property bag within the HTML output when this document is saved as a web page.

[*Example*: Consider a WordprocessingML document which contains the following content:

Stock symbol: MSFT

Date: 7/4/2006

This document might typically write out the following HTML content:

```
<p>Stock symbol: <st1:stockticker>MSFT</st1:stockticker></p>
<p>Date: <st1:date ls="trans" Month="7" Day="4"
Year="2006"">7/4/2006</st1:date></p>
```

However, if the WordprocessingML document also contains the following content within the web settings part:

```
<w:webSettings>
  <w:saveSmartTagsAsXml w:val="true" />
</w:webSettings>
```

The saveSmartTagsAsXml element specifies that all smart tags in the document must also be saved into an XML property bag at the header of the file, for example:

```
<head>
  …
  <xml>
    <o:DocumentSmartTags>
      <st1:stockticker>MSFT</st1:stockticker>
      <st1:date ls="trans" Month="7" Day="4" Year="2006">7/4/2006</st1:date>
    </o:DocumentSmartTags>
  </xml>
  …
</head>
```

This header information is in addition to the normal HTML output. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.2.37 scrollbar (Scrollbar Display Option)

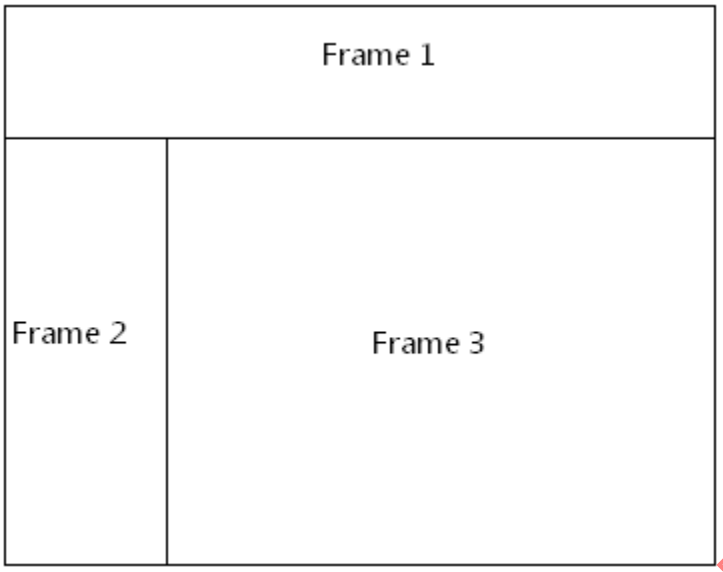
This element specifies when a scrollbar shall be visible for the contents of the current frame. When this element is set, the val attribute determines exactly when the scrollbar shall be visible. This property is analogous to the `scrolling` attribute on the `frame` element in HTML.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, the scrollbar shall only be displayed when the contents of the frame exceed the visible space for the frame (i.e. when the scrollbar is needed to display all of the content).

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frameset>
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:scrollbar w:val="auto" />
    </w:frame>
    …
  </w:frameset>
</w:frameset>
```

The scrollbar element has a val attribute of auto, which specifies that the frame must only display a scrollbar when it is needed to display all of its content. *end example*]

Attributes	Description
val (Scrollbar Display Option Value)	Specifies the criteria under which a scrollbar shall be displayed along with the contents of this frameset, as defined by the simple type referenced below. [*Example*: Consider a frameset definition within a WordprocessingML document which

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	defines the following scrollbar visibility setting:
	
	`<w:frame>`
	`  <w:scrollbar w:val="on" />`
	`  …`
	`</w:frame>`
	
	The val attribute value of on specifies that the scrollbar must always be displayed, even when it is not needed (i.e. when it would be displayed disabled). *end example*]
	
	The possible values for this attribute are defined by the ST_FrameScrollbar simple type (§17.18.32).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FrameScrollbar) is located in §A.1. *end note*]

17.15.2.38 sourceFileName (Source File for Frame)

This element specifies the ID for the relationship which specifies the source file for a single frame within a frameset document.

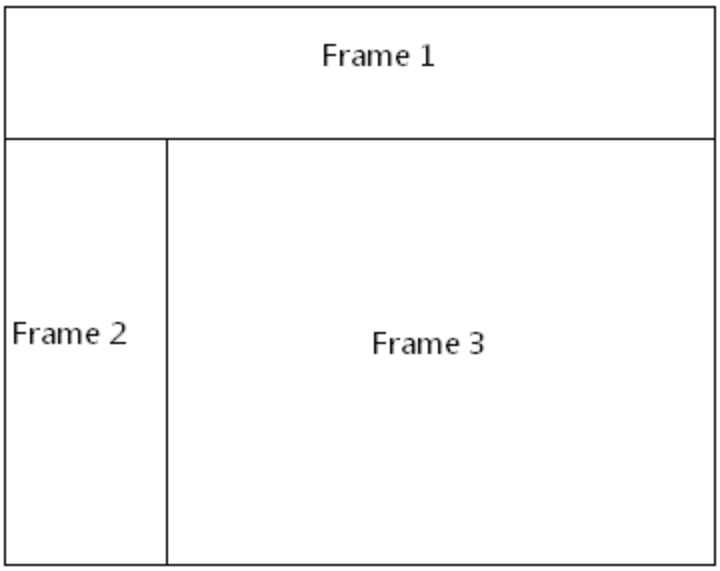
The relationship referenced by this element's id attribute shall exist in the relationship part item for the Web Settings part,or this document shall be considered non-conformant. Also, the type of the relationship referenced by this element's id attribute shall be http://purl.oclc.org/ooxml/officeDocument/relationships/frame, or this document shall be considered non-conformant.

If this element is omitted, then no source file is present for the current frame, and one can be created dynamically as needed to display content within the frame.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frameset>
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:sourceFileName r:id="rId5" />
    </w:frame>
    …
  </w:frameset>
</w:frameset>
```

The sourceFileName element specifies that the contents of this frame must be the contents of the file targeted by the relationship with ID `rId5` in the web settings part's relationship part item. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	• http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.15.2.39 sz (Frame Size)

This element specifies the size for a single frame within a frameset.

This size shall be interpreted based on the contents of the frameLayout element (§17.15.2.17) for the parent frameset, as follows:

- If the val attribute on that element is `cols`, then this element specifies the width of the frame
- If the val attribute on that element is `rows`, then this element specifies the height of the frame

Once the axis of this measurement has been established using the criteria above, the actual value of the measurement shall be determined by the following:

- If the val attribute ends in a asterisk (*), then this measurement is a relative measurement (relative to all other frames in this frameset).
- If the val attribute ends in a percentage symbol (%), then this measurement is a percentage of the height and/or width of the parent window, respectively.
- Otherwise, the value of the val attribute specifies the size of the frame in pixels. This measurement shall be interpreted in the context of the pixelsPerInch element (§17.15.2.34) to determine the width of the resulting measurement in inches.

If this element is omitted, then no information shall be implied about the size of the current frame.

[*Example*: Consider a frameset consisting of the following three frames:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The following properties define the presentation of the top frame within this frameset:

```
<w:frameset>
  …
  <w:frame>
    <w:sz w:val="300" />
    <w:name w:val="Frame 1" />
  </w:frame>
  …
  <w:pixelsPerInch w:val="150" />
</w:frameset>
```

The sz element's val attribute specifies that the size of this frame is `300` - which translates to a height of exactly 300 pixels tall. In addition, this document specifies that the intended number of pixels per inch for this measurement is `150`, resulting in a 2" tall frame height. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.2.40 sz (Nested Frameset Size)

This element specifies the size for a frameset that has been nested within another frameset. If this size appears on a root frameset, then it can be ignored and the main frameset shall encompass the entire window.

This size shall be interpreted based on the contents of the frameLayout element (§17.15.2.17) for the parent frameset (not the current nested frameset), as follows:

- If the val attribute on that element is `cols`, then this element specifies the width of the frameset
- If the val attribute on that element is `rows`, then this element specifies the height of the frameset

Once the axis of this measurement has been established using the criteria above, the actual value of the measurement shall be determined by the following:

- If the val attribute ends in a asterisk (*), then this measurement is a relative measurement (relative to all other frames in this frameset).
- If the val attribute ends in a percentage symbol (%), then this measurement is a percentage of the height and/or width of the parent frameset, respectively.
- Otherwise, the value of the val attribute specifies the size of the frameset in pixels. This measurement shall be interpreted in the context of the pixelsPerInch element (§17.15.2.34) to determine the width of the resulting measurement in inches.

If this element is omitted, then no information shall be implied about the size of the current frameset.

[*Example*: Consider a nested frameset defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:frameset>
  …
  <w:frameset>
    <w:sz w:val="50%" />
    …
  </w:frameset>
  …
  <w:pixelsPerInch w:val="150" />
</w:frameset>
```

The sz element's val attribute specifies that the size of this nested frameset is `50%` - which translates to a width of fifty percent of the width of the parent frameset's height. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.15.2.41 targetScreenSz (Target Screen Size for Web Page)

This element specifies the ideal minimum target screen size (width by height, specified in pixels) on which web pages generated when saving this document is displayed. This setting can be used to optimize the output of web pages produced from this document.

If this element is omitted, then the target screen size for web pages produced from this document shall be assumed to be `800x600`.

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:targetScreenSz w:val="1600x1200" />
</w:webSettings>
```

The targetScreenSz element's val attribute has a value of `1600x1200`, which specifies that a target screen size of 1600 by 1200 pixels must be assumed when saving this document as a web page. *end example*]

Attributes	Description

val (Target Screen Size Value)	Specifies the target screen size for web pages produced by this document, as defined by the simple type referenced below.
	
	[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:
	
	`<w:webSettings>`
	`  <w:targetScreenSz w:val="1024x768" />`
	`</w:webSettings>`
	
	The val attribute has a value of `1024x768`, which specifies that a target screen size of 1024 by 768 pixels must be assumed when saving this document as a web page. *end example*]
	
	The possible values for this attribute are defined by the ST_TargetScreenSz simple type (§17.18.86).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TargetScreenSz) is located in §A.1. *end note*]

17.15.2.42 title (Frame or Frameset Title)

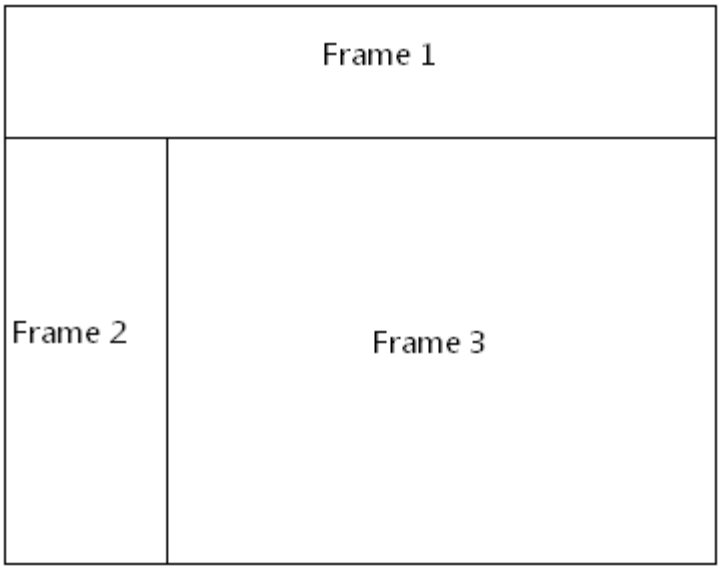
This element specifies advisory information about a single frame or frameset. The title information shall be stored in this element's val attribute. This property is analogous to the title attribute on the frame or frameset element in HTML.

If this element is omitted, then no title shall be associated with the given frame or frameset.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a WordprocessingML document that serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  <w:title w:val="Our library of documents" />
  <w:frame>
    <w:name w:val="Frame 1" />
    <w:title w:val="Menu bar" />
  </w:frame>
  <w:frameset>
    <w:title w:val="Navigation and document collection" />
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:title w:val="Navigation bar" />
    </w:frame>
    <w:frame>
      <w:name w:val="Frame 3" />
      <w:title w:val="Documents" />
    </w:frame>
  </w:frameset>
</w:frameset>
```

The title element specifies supplementary information for each frame and frameset. In this case, the frames have titles of "`Menu bar`", "`Navigation bar`", and "`Documents`", respectively, while the framesets have titles of "`Our library of documents`", and "`Navigation and document collection`", respectively. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.15.2.43 top (Top Border for HTML div)

This element specifies the border which shall be displayed at the top of the boundaries of the current HTML `div` object.

If this element is omitted, then this HTML `div` object shall not have a top border.

[*Example*: Consider a simple HTML document defined as follows:

```
<html>
  <body>
    <div style=" border-left-style:solid; border-right-style:groove; border-
right-width:1px; border-top-style:dashed; border-top-width:3px; border-bottom-
style:outset; border-bottom-width:3px">
      <p>paragraph of text</p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    </div>
  </body>
</html>
```

This HTML would therefore normally appear as follows (image scaled appropriately):

Now, when this document is saved in the WordprocessingML format, the information stored on the `div` elements is stored in the web setting part as follows:

```
<w:divs>
  <w:div w:id="1785730240">
    …
    <w:divBdr>
      <w:top w:val="dashed" w:sz="18" w:space="7" w:color="auto" />
      <w:left w:val="single" w:sz="24" w:space="4" w:color="auto" />
      <w:bottom w:val="outset" w:sz="18" w:color="auto" />
      <w:right w:val="threeDEngrave" w:sz="6" w:color="auto" />
    </w:divBdr>
  </w:div>
</w:divs>
```

The top element specifies border information about the top border for the single HTML `div` structure in the document; in this case, a 2.25 point bottom border of type `dashed`. The initial 3 pixel border was converted to 2.25 points using the following logic:

$$3\text{px} * \frac{1\text{ inch}}{96\text{ px}} * \frac{576\text{ eighth points}}{1\text{ inch}} = 18\text{ eighth points (2.25 points)}$$

end example]

This element's content model is defined by the common border properties definition in §17.3.4.

17.15.2.44 w (Frameset Splitter Width)

This element specifies the width of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and can be ignored for all nested framesets in this document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, then the default width of the splitters in this document shall be 4.5 points (90 twentieths of a point) wide. If the noBorder element (§17.15.2.31) is also specified, then this element shall be ignored and no splitters shall be displayed.

[*Example*: Consider a frameset consisting of the following three frames:

The following properties define the presentation of the splitter bars within this frameset:

```
<w:frameset>
  <w:framesetSplitbar>
    <w:w w:val="200" />
    <w:color w:val="0000FF" />
  </w:framesetSplitbar>
  …
</w:frameset>
```

The w element's val attribute specifies that the splitter must be 10 points (200 twentieths of a point) wide when the contents of this document are displayed. *end example*]

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML element with a val attribute containing a positive measurement in twentieths of a point: `<… w:val="720" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The val attribute has a value of 720, specifying that this measurement value is 720 twentieths of a point (0.5"). This value is interpreted by the parent element as needed. *end example*] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.1. *end note*]

17.15.2.45 webSettings (Web Page Settings)

This element specifies the set of web page settings that have been specified for a single WordprocessingML document. This element is the root element for the Web Settings part within a WordprocessingML document.

[*Example*: Consider the following WordprocessingML fragment for the web page settings in a WordprocessingML document:

```
<w:webSettings>
  …
  <w:frameset>
    …
  </w:frameset>
  <w:doNotUseLongFileNames w:val="true" />
  …
</w:webSettings>
```

The webSettings element contains all of the web page settings for this document. In this case, the web page settings specified for this document are: a frameset defined using the frameset element (§17.15.2.19); and a setting specifying that when this file is saved as a web page, all resulting files must not exceed 8 octets with 3 octet extension using the doNotUseLongFileNames element (§17.15.2.13). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebSettings) is located in §A.1. *end note*]

17.15.3 Language Compatibility Settings

The last group of settings in WordprocessingML are language compatibility settings. *Language compatibility settings* are optional settings used to specify changes appropriate to a subset of languages, but not usually appropriate in others. [*Example*: The doNotLeaveBackslashAlone setting changes the visual appearance of a specific character to match user expectation based on a historical use of that character in some code pages – users who have used those code pages would expect one value; those who have not would expect another. *end example*]. The behavior of each setting is fully defined in this subclause.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If language compatibility settings are needed, they are stored in the Document Settings part.

[*Note*: Although these settings can be applied in any WordprocessingML document, they are often applied when the document is created in one of the following contexts.

In the case of `ja-JP`, `ko-KR`, `zh-CN`, `zh-SG`, `zh-TW`, `zh-HK`, `zh-MO`, `ii-CN`:

- doNotLeaveBackslashAlone
- doNotExpandShiftReturn
- balanceSingleByteDoubleByteWidth
- adjustLineHeightInTable
- ulTrailSpace
- spaceForUL

In the case of `th-TH`, `lo-LA`, `km-KH`, `bo-CN`, `hy-AM`:

- applyBreakingRules

end note]

17.15.3.1 adjustLineHeightInTable (Add Document Grid Line Pitch To Lines in Table Cells)

This element specifies whether a document grid defined using the docGrid element (§17.6.5) that specifies a line grid (manually adding additional pitch to each line in the section) shall also be applied to lines within table cells in this section.

Typically, when additional line pitch is added to all lines in a section via the document grid, it is not applied to text in tables. This element, when present with a val attribute value of `true` (or equivalent), specifies that additional line pitch shall be added to lines in table cells.

[*Example*: Consider a WordprocessingML document with a single section, whose document grid is defined such that 25.9 points of additional line pitch are added to each line in the section, as follows:

```
<w:docGrid w:type="lines" w:linePitch="518"/>
```

If text was entered into this section, the default behavior would have line pitch only added to lines which are not in a table cell:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid.

This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid.

This section has a character grid. This section has a character grid. This section has a character grid.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:adjustLineHeightInTable />
</w:compat>
```

Then all lines in this document would have the line pitch from the document grid added to them, resulting in the following output:

This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid.

This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid. This section has a character grid.

This section has a character grid. This section has a character grid. This section has a character grid.

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.2 applyBreakingRules (Use Legacy Ethiopic and Amharic Line Breaking Rules)

This element specifies whether applications shall use a legacy set of line breaking rules when determining line breaks for text consisting of Ethiopic and/or Amharic characters.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Typically, when line breaking this text, applications should allow line breaks to occur after a character between the UTF-16 (hexadecimal) values 0x1361 and 0x1368 when those characters appear in the document's content. This element, when present with a val attribute value of `true` (or equivalent), specifies that when a line break would occur after a character between the UTF-16 hexadecimal) values 0x1361 and 0x1368, the line break shall occur before all instances of these characters (i.e. no break opportunity shall be afforded after a character in this range).

[*Example*: Consider a WordprocessingML document with a series of Ethopic characters in this range. The default presentation would have any line breaks pushed before or after these characters, ensuring that the characters remain together on a single line.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:applyBreakingRules />
</w:compat>
```

Then a line break opportunity must be afforded at any point in a range of these characters, as needed. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.3 balanceSingleByteDoubleByteWidth (Balance Single Byte and Double Byte Characters)

This element specifies whether applications shall balance the width of Single Byte Character Set characters and Double Byte Character Set characters when rendering WordprocessingML documents. Specifically, this element specifies to adjust the fixed pitch fonts' half-width space character and full-width space character to attain a 1 to 2 ratio.

[*Note*: This element is used with East Asian content. Layout and line breaking for East Asian text is dependent on the character width. Half width characters (or Hankaku characters) are one half of an em wide, and full width characters (or Zenkaku characters) are one em wide. Legacy encoding often used a single byte to encode half-width characters and two bytes to encode full width characters. *end note*]

Typically, no adjustment is done on any character when it is displayed as part of a WordprocessingML document. This element, when present with a val attribute value of `true` (or equivalent), specifies that character sizes shall be adjusted as needed to meet the 1:2 ratio described above.

[*Example*: Consider a WordprocessingML document with both SBCS and DBCS characters. The default presentation would have the text displayed as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:balanceSingleByteDoubleByteWidth />
</w:compat>
```

Then this character-level adjustment must be performed, resulting in the following output:

This adjustment is usually very minute in nature, therefore the result is better illustrated by showing how the characters after the English text were pushed out due to the width balancing of that text:

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.4 compatSetting (Custom Compatibility Setting)

This element specifies a custom compatibility setting. The semantics for this element are implementation-defined. [*Note*: This element can be used to store the transitional compatibility settings specified in Part 4 of ISO/IEC 29500. *end note*]

[*Example*: Consider the following markup:

```
<w:compatSetting w:name="cooper" w:uri="http://www.example.com/exampleSetting"
w:val="1"/>
```

This custom compatibility setting specifies that:

- It is defined under the `http://www.example.com/exampleSetting` namespace.
- Its name is `cooper`.
- Its value is `1`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Beyond this, the properties of this setting are implementation-defined by the creator of the http://www.example.com/exampleSetting namespace. *end example*]

Attributes	Description
name (Name of Setting)	Specifies the name of a custom compatibility setting. [*Example*: Consider the following markup: `<w:compatSetting w:name="cooper" w:uri="http://www.example.com/exampleSetting" w:val="1"/>` The name of this custom compatibility setting is `cooper`. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
uri (Namespace of Setting)	Specifies the namespace under which the compatibility setting is defined. [*Example*: Consider the following markup: `<w:compatSetting w:name="cooper" w:uri="http://www.example.com/exampleSetting" w:val="1"/>` The namespace of this custom compatibility setting is http://www.example.com/exampleSetting. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
val (Value of Setting)	Specifies the value of a custom compatibility setting. [*Example*: Consider the following markup: `<w:compatSetting w:name="cooper" w:uri="http://www.example.com/exampleSetting" w:val="1"/>` The value of this custom compatibility setting is 1. This value is interpreted using the implementation-defined behavior published by the creator of this property. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CompatSetting) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.15.3.5 doNotExpandShiftReturn (Don't Justify Lines Ending in Soft Line Break)

This element specifies whether applications should fully justify the contents of incomplete lines which end in a soft line break when the parent paragraph is fully justified using the jc element (§17.3.1.13).

Typically, applications shall fully justify all lines in a paragraph when that setting is specified using the jc element except for the last line in the paragraph (the line ending with the paragraph mark). This element, when present with a val attribute value of `true` (or equivalent), specifies that any line which ends in a soft line break shall also not be fully justified when the paragraph specifies that setting.

[*Example*: Consider a WordprocessingML document with a paragraph whose first single line consists of East Asian characters followed by a soft paragraph mark. The default presentation would have the contents of that line fully justified:

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotExpandShiftReturn />
</w:compat>
```

Then this line is not fully justified, as it ends with a soft line break, resulting in the following output:

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.6 doNotLeaveBackslashAlone (Display Backslash As Yen Sign)

This element specifies whether applications should autodisplay the backslash character using the yen character when displaying the contents of this document.

Typically, no automatic display-only conversion of one character to another is performed. This element, when present with a val attribute value of `true` (or equivalent), specifies that all occurances of the backslash character (\, U+005C) shall automatically be displayed using the yen symbol (¥, U+00A5) when the contents of the document are displayed. This setting does not change the Unicode value of the character stored in the underlying WordprocessingML document.

[*Rationale*: In Japanese code page 932, 0x5C is the yen sign (whereas, in most other code pages, it is the reverse solidus–also known as the backslash). In order to accommodate the user expectation that this code point appear as the yen sign, this setting dictates that the character be remapped, for display only, to the Unicode character ¥, such that the expected appearance is maintained. *end rationale*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a WordprocessingML document containing the following:

```
Hello \ world.
```

The default presentation would have exactly that:

```
Hello \ world.
```

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:doNotLeaveBackslashAlone />
</w:compat>
```

Then the backslash would be displayed as ¥, resulting in the following output:

```
Hello ¥ world.
```

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.7 spaceForUL (Add Additional Space Below Baseline For Underlined East Asian Text)

This element specifies whether East Asian content in a WordprocessingML document which has been underlined using the u element shall have additional descent added to the properties of the font in order to ensure that there is adequate spacing between the characters in the font and the underlining applied to the text.

Typically, no adjustments are made to the contents of text runs containing East Asian text which have been underlined. This element, when present with a val attribute value of `true` (or equivalent), specifies that whenever the following conditions are met:

- The text run contains East Asian characters
- The text run is not using baseline font alignment as defined by the textAlignment property

That the larger of the following two values is added to the descent property of that font in order to provide additional padding between the text characters and the underline:

- 3 percent of the font size
- 40 twentieths of a point (31 twentieths of a point for Japanese text)

[*Example*: Consider a WordprocessingML document consisting of a single run of underlined Japanese text, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:rPr>
      <w:u w:val="double" />
    </w:rPr>
    <w:t>クリスタ</w:t>
  </w:r>
</w:p>
```

If this document is displayed, then the text is laid out along with the underline, as follows:

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:spaceForUL />
</w:compat>
```

Then the additional descent specified using the logic above is added to the text, resulting in the following output:

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.15.3.8 ulTrailSpace (Underline All Trailing Spaces)

This element specifies whether applications shall display underlining beneath all trailing spaces in the contents of a line when those contents are underlined. *Trailing spaces* are all space characters which are not followed by non-space characters on the same line.

Typically, applications do not display underlining on all trailing spaces which have the underline property applied to them. This element, when present with a val attribute value of `true` (or equivalent), specifies that all characters with underline applied, including trailing spaces, shall display underlining if it is applied to that content.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider a WordprocessingML document with the following line of Latin alphabetical character and punctuation, trailed by a series of spaces:

```
<w:r>
  <w:rPr>
    <w:u w:val="single"/>
  </w:rPr>
  <w:t>Example text. Example text. Example text. Example text. Example text.
</w:t>
</w:r>
```

The default presentation would have no underlining on those trailing spaces:

Example text. Example text. Example text. Example text. Example text.

However, if this compatibility setting is turned on:

```
<w:compat>
  <w:ulTrailSpace />
</w:compat>
```

Then all trailing spaces would be underlined, resulting in the following output:

Example text. Example text. Example text. Example text. Example text.

end example]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16 Fields and Hyperlinks

Most text in a word processing document is static; that is, unless it is directly changed as the result of editing, its contents remain the same, no matter how the rest of the document might change. However, certain useful pieces of information can change value over the life of a document. Consider the case of a reference to a page number, as in "For more information on this topic, see page 56." Clearly, hard coding the page number as 56 means that that number needs to be manually replaced as the document's size or layout is changed. Even a simple change to any margin, line spacing, or font size can invalidate such references.

Fields provide a mechanism for placeholders, such as page reference numbers, that can be added to a document such that those placeholders are replaced by their corresponding values when the document is rendered for display or print. Other applications for fields include, but are not limited to, automatic numbering of tables and figures, document creation and current date and time, document author information, and the computation of totals for a table column.

A *field* is a set of codes that instructs a WordprocessingML consumer to insert text, graphics, page numbers, and other material into a document automatically. [*Example*: The DATE field causes the current date to be inserted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example] The text or graphics inserted into a document when a consumer carries out a field's codes is referred to as the *field result* (or *field value*) for that field. The default value for a field result is an empty string. The act of carrying out a field's codes is referred to as a *field update*. A field update shall be carried out such that the content contained within the field result correctly reflects the intended semantics of the field codes (as defined in the subclauses of §17.16.5). For a simple field defined via the fldSimple element (§17.16.19), this means replacing its child elements with the appropriate WordprocessingML content; for a complex field represented by use of properly matched fldChar elements (§17.16.18), this means replacing any content between the fldChar element with an fldCharType attribute value of `separate` and an fldCharType attribute value of end. As to when any field is updated is outside the scope of ISO/IEC 29500.

17.16.1 Syntax

The syntax rules in this subclause follow the system shown in ISO/IEC 14977: literal text is surrounded by double-quotes (or by apostrophes); the left-square-bracket and right-square-bracket designate the start and end of an option; the left-curly-bracket and right-curly-bracket designate the start and end of a sequence of one-or-more items; the vertical-line indicates an alternative; and each rule ends with a semicolon. Whenever hyphen is used as the exception-symbol (as per ISO/IEC 14977), it is surrounded by white space, and further clarified by a comment.

The syntax rules below were derived from the field-specific rules in the detailed definitions shown in §17.16.5.*. [*Note*: In order to produce an automated verifier, field-specific rules in subclauses 17.16.5.* must also be considered. *end note*]

When used in narrative, production names are set in an italic style, as in *comparison*, *field-argument*, and *switches*.

The syntax of a field is as follows:

```
field=
   field-type,  [switches]  ;
field-type=
   date-and-time  |
   document-automation  |
   document-information  |
   document-property  |
   equations-and-formulas  |
   index-and-tables  |
   links-and-references  |
   mail-merge  |
   numbering  |
   user-information  |
   form-field  |  user-defined-field  ;
user-defined-field=
   letter,  {letter}  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
date-and-time=
   "CREATEDATE"  |  "DATE"  |  "EDITTIME"  |  "PRINTDATE"  |
   "SAVEDATE"  |  "TIME"  ;
document-automation=
   "COMPARE",  comparison  |
   "DOCVARIABLE",  field-argument  |
   "GOTOBUTTON",  2 * field-argument  |
   "IF",  comparison,  2 * field-argument  |
   "MACROBUTTON",  2 * field-argument  |
   "PRINT",  field-argument  |
   "PRIVATE"  ;
document-information=
   "FILENAME"  |  "FILESIZE"  |  "LASTSAVEDBY"  |  "NUMCHARS"  |
   "NUMPAGES"  |  "NUMWORDS"  |  "TEMPLATE"  ;
document-property=
   "AUTHOR",  [field-argument]  |
   "COMMENTS",  [field-argument]  |
   "DOCPROPERTY",  docprop-category  |
   "KEYWORDS",  [field-argument]  |
   "SUBJECT",  [field-argument]  |
   "TITLE",  [field-argument]  ;
equations-and-formulas=
   "=",  expression  |
   "ADVANCE"  |
   "SYMBOL",  field-argument  ;
index-and-tables=
   "INDEX"  |
   "RD",  field-argument  |
   "TA"  |
   "TC",  field-argument  |
   "TOA"  |
   "TOC"  |
   "XE",  field-argument  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
links-and-references=
   "AUTOTEXT",  field-argument  |
   "AUTOTEXTLIST",  field-argument  |
   "BIBLIOGRAPHY"  |
   "CITATION",  field-argument  |
   "HYPERLINK",  field-argument  |
   "INCLUDEPICTURE",  field-argument  |
   "INCLUDETEXT",  field-argument,  [field-argument]  |
   "LINK",  2 * field-argument,  [field-argument]  |
   "NOTEREF",  field-argument  |
   "PAGEREF",  field-argument  |
   "QUOTE",  field-argument  |
   ["REF"],  field-argument  |  (* see §17.16.5.51 *)
   "STYLEREF",  field-argument  ;
mail-merge=
   "ADDRESSBLOCK"  |
   "ASK",  2 * field-argument  |
   "DATABASE"  |
   "FILLIN",  [field-argument]  |
   "GREETINGLINE"  |
   "MERGEFIELD",  field-argument  |
   "MERGEREC"  |
   "MERGESEQ"  |
   "NEXT"  |
   "NEXTIF",  comparison  |
   "SET",  2 * field-argument  |
   "SKIPIF",  comparison  ;
numbering=
   "LISTNUM",  [field-argument]  |
   "PAGE"  |
   "REVNUM"  |
   "SECTION"  |
   "SECTIONPAGES"  |
   "SEQ",  identifier,  [field-argument]  ;
user-information=
   "USERADDRESS",  [field-argument]  |
   "USERINITIALS",  [field-argument]  |
   "USERNAME",  [field-argument]  ;
form-field=
   "FORMCHECKBOX"  |  "FORMDROPDOWN"  |  "FORMTEXT"  ;
bookmark-name=
   identifier  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
cell-name=
   column-name,  row-name  ;
cell-range=
   cell-name,  colon,  cell-name  |
   row-name,  colon,  row-name  |
   column-name,  colon,  column-name  ;
cell-reference=
   cell-name  |
   cell-name,  {  comma,  cell-name  }  |
   cell-range  ;
character=
  as defined by the production Char in the XML 1.0 specification, §2.2.
colon=
   ":"  ;
column-name=
   letter  +  [{letter}]  ;   (* allowing for A, … Z, AA, …., ZZ, AAA, … column
naming *)
comma=
   ","    ;
comparison=
   expression,  comparison-operator,  expression  ;
comparison-operator=
   "="  |  "<>"  |  "<"  |  "<="  |  ">"  |  ">="  ;
constant=
   number  ;
date-and-time-formatting-switch=
   "\@",  switch-argument  (* as defined in §17.16.4.1 *) ;
docprop-category=
   "AUTHOR"  |  "BYTES"  |  "CATEGORY"  |  "CHARACTERS"  |
   "CHARACTERSWITHSPACES"  |  "COMMENTS"  |
   "COMPANY  |  CREATETIME"  |  "HYPERLINKBASE"  |
   "KEYWORDS"  |  "LASTPRINTED"  |  "LASTSAVEDBY"  |
   "LASTSAVEDTIME"  |  "LINES"  |  "MANAGER"  |
   "NAMEOFAPPLICATION"  |  "ODMADOCID"  |  "PAGES"  |
   "PARAGRAPHS"  |  "REVISIONNUMBER"|  "SECURITY"  |
   "SUBJECT"  |  "TEMPLATE"  |  "TITLE"  |
   "TOTALEDITINGTIME"  |  "WORDS"  ;
double-quote=
   '""'   ;  (* one double-quote character *)
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
expression=
   "(", expression, ")" |
   comparison |
   constant |
   prefix-operator, expression |
   expression, infix-operator, expression |
   field |
   bookmark-name, [cell-reference] |
   function |
   cell-reference |
   text |
   double-quote, text, double-quote ;
field-argument=
   text |
   double-quote, text, double-quote ;
field-specific-switch=
   "\", character, [character], [field-argument] ;
      (* no whitespace is permitted after the backslash,
      also see definition of each field in §17.16.5 *)
formatting-switch=
   date-and-time-formatting-switch |
   numeric-formatting-switch |
   general-formatting-switch ;
fractional-part=
   decimal-digit {decimal-digit} ;
full-stop=
   "."    ; (* also known as “period” *)
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
function=
   "ABS(", expression, ")" |
   "AND(", expression, "," expression, ")" |
   "AVERAGE(", list, ")" |
   "COUNT(", list, ")" |
   "DEFINED(", expression, ")" |
   "FALSE" |
   "INT(", expression, ")" |
   "MAX(", list, ")" |
   "MIN(", list, ")" |
   "MOD(", expression, ",", expression, ")" |
   "NOT(", expression, ")" |
   "OR(", expression, ",", expression, ")" |
   "PRODUCT(", list, ")" |
   "ROUND(", expression, ",", expression, ")" |
   "SIGN(", expression, ")" |
   "SUM(", list, ")" |
   "TRUE" ;
general-formatting-switch=
   "\*", switch-argument ; (* as defined in §17.16.4.3 *)
identifier=
   character + [{character}] ;
infix-operator=
   "-" | "^" | "*" | "/" | "%" | "+" | "-" | "=" |
   "<>" | "<" | "<=" | ">" | ">=" ; (* see §17.16.3.3 *)
letter=
  "a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
  "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
  "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
  "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z" ;
list=
   expression, {list-separator, expression} ;
list-separator=
   comma | semicolon ;
      (* depending on the document's listSeparator, §17.15.1.56*)
number=
   whole-number-part, [full-stop] |
   full-stop, fractional-part |
   whole-number-part, full-stop, fractional-part ;
numeric-formatting-switch=
   "\#", switch-argument ;       (* as defined in §17.16.4.2 *)
prefix-operator=
   "-" ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
row-name=
   whole-number-part  ;
semicolon=
   ";"    ;
switch-argument=
   text  |
   double-quote,  text,  double-quote  ;
switches=
   {field-specific-switch},  [formatting-switch]  ;
text=
   character,  {character}  ;
whole-number-part=
   decimal-digit,  {decimal-digit}  ;
```

Formulas (also called expressions) are discussed in §17.16.3, constants are discussed in §17.16.3.1, bookmarks are discussed in §17.16.3.2, operators are discussed in §17.16.3.3, functions are discussed in §17.16.3.4, table cell references are discussed in §17.16.3.5, and formatting switches are discussed in §17.16.4.

The semantics of a field having a *user-defined-field type* are unspecified.

If the *text* in a *field-argument* or *switch-argument* contains white space, the delimiting double-quote characters shall be present; otherwise, they are optional. To include a double-quote character in *text*, it shall be preceded with a backslash (\). [*Example*: The field argument `"\"name\""` results in the argument's actually being `"name"`. *end example*] To include a backslash character in *text*, it shall be preceded with another backslash (\). [*Example*: File system pathnames on some systems use a backslash as a directory separator, as in the field

```
INCLUDETEXT "E:\\ReadMe.txt"
```

in which case, each such separator needs to be preceded with a backslash, as shown above. *end example*]

Arbitrary amount of white space can occur before the first token, after the last token, and between successive tokens, including no white space at all, except that in a *field-specific-switch*, no white space is permitted between the initial "\" token and the *character* that follows it.

[*Example*: Here are examples of some fields:

```
DATE
DATE \@ "dddd, MMMM dd, yyyy"
DATE \@ "dddd, MMMM dd, yyyy" \h
```

The field result of all three is today's date: The first field uses some implementation-defined format and the Gregorian calendar; the second field uses the specified format and the Gregorian calendar; and the third field uses the specified format and the Hijri lunar calendar. When rendered in a US-English context on December 31, 2005, the results of these fields were as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
12/31/2005
Saturday, December 31, 2005
AsSabt, Thoul Ki'dah 30, 1426
```

end example]

The names of fields are alphabetic tokens [*Example*: Some field-type names are ASK, COMMENTS, NEXT, and SET. *end example*]. These tokens are called *field-type names*. Field-type names are case-insensitive. [*Example*: The field-type names DATE, Date, dAtE, and date are equivalent. *end example*]

The *characters* immediately following the "\" in *field-specific-switch*are case-insensitive. [*Example*: \b and \B are equivalent. *end example*]

There is no ordering of *field-specific-switch* entries in *switches*.

17.16.2 XML representation

Fields shall be implemented in XML using either of two approaches:

- As a *simple field implementation*, using the fldSimple element, or
- As a *complex field implementation*, using a set of runs involving the fldChar and instrText elements.

For a simple field implementation, only one element, fldSimple, shall be used, in which case, its instr attribute shall contain a *field*, and the body of the element shall contain the most recently updated field result. [*Example*: Here is the corresponding XML for a simple field implementation of DATE:

```
<w:fldSimple w:instr="DATE">
  <w:r>
    <w:t>12/31/2005</w:t>
  </w:r>
</w:fldSimple>
```

end example]

For a complex field implementation, a set of runs shall be used with each run containing, in sequence, the following elements:

- fldChar with attribute fldCharType value begin,
- One or more instrText elements, which, collectively, contain a complete *field*,
- Optionally,
  - fldChar with attribute fldCharType value separate, which separates the field from its field result,
  - Any number of runs and paragraphs that contains the most recently updated field result, and
- fldChar with attribute fldCharType value end.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: Fields that are for display purposes only have no need to, and do not, store a field result. *end note*][*Example*: Here is the corresponding XML for a complex field implementation of DATE:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> DATE </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r>
  <w:t>12/31/2005</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

end example]

[*Note*: Every simple field implementation for a given field has a corresponding complex field implementation. However, not every complex field implementation has a corresponding simple field implementation. If some characters in a *field* have different run properties than others, that field must be implemented using multiple runs, and that requires that complex field implementation be used. For an example, see §17.16.4.3, where the first letter of a DATE field is made bold, underlined, and red, while the other letters have none of these properties. *end note*]

As shown in §17.16.1, the *instruction* of one *field* can be another *field*, allowing fields to nest. In such cases, the XML run sequence for the inner field is defined at the point of reference for that inner field, inside the outer field's XML run sequence. [*Example*: Consider the following sentence:

It's `IF DATE \@ "M-d"<>"1-1" "not "` new year's day.

The `IF` field contains the nested field `DATE \@ "M-d"`. When updated, on January 1 of any year, the result sentence is "It's new year's day." On all other days of the year, the resulting sentence is "It's not new year's day."

Here is one way of writing the corresponding XML:

```
<w:r>
  <w:t xml:space="preserve">It’s </w:t>
</w:r>
<w:r …>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:instrText xml:space="preserve">IF </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> DATE  \@ "M-d"  </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:instrText>1-4</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
<w:r>
  <w:instrText>&lt;&gt;"1-1" "not "</w:instrText>
</w:r>
<w:r …>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:t xml:space="preserve">not </w:t>
</w:r>
<w:r …>
  <w:fldChar w:fldCharType="end"/>
</w:r>
<w:r>
  <w:t>new year’s day!</w:t>
</w:r>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.3 Formulas and expressions

A field instruction can involve a calculation via a *formula*, which is simply an *expression* that is an arbitrary complex arithmetic expression involving constants (§17.16.3.1), bookmarks that refer to *expressions* (§17.16.3.2), arithmetic and logical operators (§17.16.3.3), functions (§17.16.3.4), values of cells in a table (§17.16.3.5), and *fields* that result in a single value. *expression* can contain grouping parentheses to document the default precedence or to override it.

 ©ISO/IEC 2016 – All rights reserved

All arithmetic terms in an *expression* are real numbers. Infinities and NaN (Not-a-Number) are not supported. [*Example*: In the expression `1/3`, although the operands appear to be integers, they are, in fact real numbers, and the result is `0.33`. *end example*]

17.16.3.1 Constants

A constant is a number. Exponents are not supported.

[*Example*: Here are some constants: `1234`, `1234.560`, `1234.`, and `.1234`. *end example*]

17.16.3.2 Bookmarks

Any arbitrary piece of text and/or graphics in a WordprocessingML document can be assigned a name, called a *bookmark*. If a bookmark references text that represents an *expression*, that bookmark's name can be used as an operand in another *expression*. If a whole field is bookmarked, its bookmark name can also be used as an operand in an *expression*. [*Example*: Given that X is a bookmark for the text 4, Y is a bookmark for the text 2, and `Result` is a bookmark for the following field:

```
=X + Y
```

the field

```
=Result * 10
```

has the result `60`. *end example*]

17.16.3.3 Operators

The *operators* permitted in *expression* are:

Operators		
Operator	Description	Precedence
-	Unary minus	highest
^	Powers and roots	
*	Multiplication	
/	Division	
%	Percentage	
+	Addition	
-	Subtraction	
=	Equal to	
<>	Not equal to	
<	Less than	lowest
<=	Less than or equal to	
>	Greater than	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Operators		
>=	Greater than or equal to	

Operators in *expression* having the same precedence associate left-to-right.

[*Example*: Given that X is a bookmark for the text 4, and Y is a bookmark for the text 2, the field

```
=((-1 + X^2) * 3 - Y)/2
```

produces the result `21.5`. *end example*]

The equality, inequality, and relational operators yield 1 for true and 0 for false. An expression with value 0 tests logically false while one with any non-zero value tests true.

17.16.3.4 Functions

A *function* is a predefined procedure that computes and returns a result. Functions defined below with a parameter list of *list* accept two or more arguments separated by commas (,) or semicolons (;). As to which separator is permitted, is defined by the document's listSeparator (§17.15.1.56) element. Arguments to functions can be constants, formulas, or bookmark names that refer to constants or formulas. The functions AVERAGE, COUNT, MAX, MIN, PRODUCT, and SUM can also accept references to table cells as arguments. In the context of a table cell, functions taking a *list* also accept a single argument that designates a named-list of contiguous cells (§17.16.3.5). Function names are not case-sensitive, and white space can occur between a function's name and its argument list, if any.

The functions supported are as follows:

Functions	
Function	Description
ABS(*x*)	Returns the absolute value of *x*.
AND(*x*,*y*)	Returns 1 if the logical expressions *x* and *y* are both true; otherwise, it returns 0.
AVERAGE(*list*)	Returns the average value of the items in *list*.
COUNT(*list*)	Returns the number of items in *list*.
DEFINED(*x*)	Returns 1 if the expression *x* is well formed; otherwise, it returns 0.
FALSE	Returns 0.
INT(*x*)	Returns the value of the integer part of *x*.
MAX(*list*)	Returns the largest value in *list*.
MIN(*list*)	Returns the smallest value in *list*.
MOD(*x*,*y*)	Returns the value *x* - *ny*, for some integer *n* such that, if *y* is nonzero, the result has the same sign as *x* and magnitude less than the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Functions	
	magnitude of *y*. If *y* is zero, a diagnostic shall be issued. (*y* need not be a whole number.) [*Example*: `MOD(21,5)` results in 1 `MOD(21,-5)` results in 1 `MOD(-21,5)` results in -1 `MOD(-21,-5)` results in -1 *end example*]
`NOT(`*x*`)`	Returns 0 if the logical expression *x* is true, or 1 if the expression is false.
`OR(`*x*`,`*y*`)`	Returns 1 if either or both logical expressions *x* and *y* are true; otherwise, it returns 0.
`PRODUCT(`*list*`)`	Returns the result of multiplying together all members in *list*.
`ROUND(`*x*`,`*y*`)`	Returns the value of *x* rounded to the specified number of decimal places indicated by floor(*y*), where floor has the mathematical meaning. If *y* is negative, any fractional part is discarded and the integer part of the value is rounded to the corresponding power of 10.
`SIGN(`*x*`)`	Returns 1 if *x* is positive; returns 0 if *x* is zero; and returns –1 if *x* is negative.
`SUM(`*list*`)`	Returns the sum of the items in *list*.
`TRUE`	Returns 1.

17.16.3.5 Table cell references

Items in a WordprocessingML table are organized into rows and columns with the box formed by the intersection of a row and column being called a *cell*. Cells have names such as A1, A2, B1, B2, and so on, with the letter representing a column and the number representing a row. The cell at the top-left corner of each table is named A1. Column letters are not case-sensitive.

A *cell reference* shall be one of the following:

- The name of a cell.
- A comma-separated set of cell names.
- A cell range where a colon (:) is used to separate the first and last cells in a designated range of cells that has a contiguous rectangular shape. Specifying a row or column's name only as the first and last cell in a range, selects that whole row or column, regardless of the number of rows and columns the table has now or might have in the future.

An *expression* inside a table's cell can have operands that are references to other cells in that table.

[*Example*: Consider a table with three rows (1, 2, and 3) and two columns (A and B):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A1 + B1	Returns the sum of the contents of cells A1 and B1.
SUM(A1,B2,A3)	Returns the sum of the contents of the list of cells.
SUM(B1:B3)	Returns the sum of the contents of all cells between B1 and B3, inclusive.
SUM(B:B)	Returns the sum of the contents of all cells in column B (even if new rows are added later).
SUM(A1:B2)	Returns the sum of the contents of all (four) cells in the rectangular grid delimited by A1 and B2, inclusive.
SUM(1:1,2:2)	Returns the sum of the contents of all cells in rows 1 and 2.

end example]

When used in a table cell, the functions taking a *list* argument can have a single argument of ABOVE, BELOW, LEFT, or RIGHT, spelled in any case combination. Such lists designate, respectively, all the cells above, below, to the left of, or to the right of that cell. However, the designated range terminates if a cell with blank or non-numeric contents is reached, except that if the first cell is blank, it is treated as containing 0. [*Example*: Given the following table:

12	=COUNT(BELOW)	
	10	
2	20	=SUM(LEFT)
3	xxx	
=AVERAGE(ABOVE)	40	

AVERAGE(ABOVE) results in 2.5, the average of cells A4 and A3; COUNT(BELOW) results in 2, B2 and B3; and SUM(LEFT) results in 22, the sum of B3 and A3. *end example*]

An *expression* used outside a table or in a cell of one table can refer to cells in a second table by making a bookmark to that second table and qualifying cell names in that table by their table name using the form

(*tableBookmarkName cellReference*)

[*Example*: Given that Table1 is a bookmark for a 3x2 table, =SUM(Table1 A1:A3) book results in the sum of column A's cells. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.4 Field formatting

The result of a field has a *format*, either by default or because that field contains a *formatting-switch*. There are three kinds of field formatting: date and time (§17.16.4.1), numeric (§17.16.4.2), and general (§17.16.4.3).

17.16.4.1 Date and time formatting

date-and-time-formatting-switch=
\@ *switch-argument* ;

©ISO/IEC 2016 – All rights reserved

A *date-and-time-formatting-switch* specifies the format of a date or time result. [*Note*: This switch is sometimes called a *picture* switch because it allows the use of symbols to represent the format of the field result. *end note*] If the result of a field is not a date or time, this switch has no effect.

If no *date-and-time-formatting-switch* is present, a date or time result is formatted in an implementation-defined manner.

A date and time *switch-argument* is made up of a series of *picture items*.

Date Formatting Picture Items	
Picture Item	Description
aaa	Formats the day of the week or month in an abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is `ja-JP` or ko-KR, display in the corresponding language. • For all other lang element values, display picture item as text. Multiple instances of the picture item create repeated content.
A	Formats the day of the month as a number without a leading zero for single-digit days in Japanese numerals. Multiple instances of the picture item create repeated content.
bb	Formats the year as a 2-digit number according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is zh-TW, zh-CN, zh-HK, zh-SG, or zh-MO, use the Gregorian year [ISO 8601] . • For all other lang element values, use Thai Buddhist Era year This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content unless the pattern contains the “bbbb” picture item. Those portions are formatted following the description outlined for the “bbbb” picture item.
bbbb	Formats the year as a 4-digit number according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ZH-TW, zh-CN, zh-HK, zh-SG, or zh-MO, use the Gregorian year [ISO 8601] • For all other lang element values, use Thai Buddhist Era year This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content.
BB	Formats the year as a 2-digit number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Defaults to the Gregorian [ISO 8601] calendar, but also responds to the \s and \h switches. Multiple instances of the picture item create repeated content as specified by the following: 3. Working from the text direction of the run, create as many groups as possible that contain “BBBB” in each group. Format each group using the description outlined for the “BBBB” picture item. 4. Repeat step 1 for groups of “BB” using the description for that picture item.
BBBB	Formats the year as a 4-digit number. Defaults to the Gregorian calendar [ISO 8601], but also responds to the \s and \h switches. Multiple instances of the picture item create repeated content as specified by the “BB” picture item description.
d	Formats the day of the week or day of the month as a number without a leading zero for single-digit days. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “dddd” in each group. Format each group using the description outlined for the “dddd” picture item. 2. Repeat step 1 for groups of “ddd”, “dd”, and “d” using the respective description for each picture item group.
dd	Formats the day of the week or day of the month as a number with a leading zero0 for single-digit days. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “d” picture item description.
ddd	Formats the day of the week or month in its abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	of the \s and \h switches, and the bb, bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ๖๖๖๖, ๖๖๖, ๖๖, and ๖ picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “d” picture item description.
dddd	Formats the day of the week as its full name according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ๖๖๖๖, ๖๖๖, ๖๖, and ๖ picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “d” picture item description.
D	Formats the day of the week or day of the month as a number without a leading zero for single-digit days. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “DDDD” in each group. Format each group using the description outlined for the “DDDD” picture item. 2. Repeat step 1 for groups of “DDD”, “DD”, and “D” using the respective description for each picture item group.
DD	Formats the day of the month as a two-digit number (with a leading zero for single-digit days). Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “D” picture item description.
DDD	Formats the day of the week in an abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ๖๖๖๖, ๖๖๖, ๖๖, and ๖ picture item (to the Thai Buddhist Era calendar).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Multiple instances of the picture item create repeated content as specified by the “D” picture item description.
DDDD	Formats the day of the week as its full name according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ๖๖๖๖, ๖๖๖, ๖๖, and ๖ picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “D” picture item description.
e	Formats the Japanese Emperor Era era with no leading zero for single-digit years. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “ee” in each group. Format each group using the description outlined for the “ee” picture item. 2. Repeat step 1 for groups of “e” using the description for that picture item.
ee	Formats the Japanese Emperor Era era with a leading zero for single-digit years. Multiple instances of the picture item create repeated content as specified by the “e” picture item description.
E	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ja-JP, display the Japanese Emperor Era era with no leading zero for single-digit years in the corresponding language. • If the lang element is zh-TW, display the Taiwanese year in the corresponding language. • For all other values, display the Gregorian year [ISO 8601] as a four-digit number using ja-JP. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “EE” in each group. Format each group using the description outlined for the “EE” picture item. 2. Repeat step 1 for groups of “E” using the description for that picture item.
EE	Formats the Gregorian year [ISO 8601] as a four-digit number, according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	• If the lang element is ja-JP, zh-TW, zh-CN, zh-HK, zh-SG, or zh-MO display in the corresponding language. • For all other lang element values, display in ja-JP. Multiple instances of the picture item create repeated content as specified by the “E” picture item description.
g	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: If the lang element is ko-KR or zh-TW, display nothing. For all other lang element values, display Japanese Emperor Era era as its abbreviated form in en-US. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “ggg” in each group. Format each group using the description outlined for the “ggg” picture item. 2. From the remaining values, repeat step 1 for groups of “gg” using the description for that picture item. 3. From the remaining values, repeat step 1 for groups of “g” using the description for that picture item.
gg	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ja-JP, display the Japanese Emperor Era era as its abbreviated form in the corresponding language. • If the lang element is ko-KR, display the Korean Tangun era as its full name in the corresponding language. • If the lang element is zh-TW, display the Taiwanese era as its full name in the corresponding language. • For all other lang element values, display the Japanese Emperor Era era as its abbreviated form in ja-JP. Multiple instances of the picture item create repeated content as specified by the “g” picture item description.
ggg	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ja-JP, display the Japanese Emperor Era era as its full name in the corresponding language. • If the lang element is ko-KR, display the Korean Tangun era as its full name in the corresponding language. • If the lang element is zh-TW, display the Taiwanese era as its full name in the corresponding language. • For all other lang element values, display the Japanese Emperor Era era as its full name in ja-JP.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Multiple instances of the picture item create repeated content as specified by the “g” picture item description.
G	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. • If the lang element is ko-KR or zh-TW, display nothing. • For all other lang element values, display the Japanese Emperor Era era as its abbreviated form in ja-JP. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “GG” in each group. Format each group using the description outlined for the “GG” picture item. 2. Repeat step 1 for groups of “G” using the description for that picture item.
GG	Formats the era according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. • If the lang element is ko-KR, display the Korean Tangun era as its full name in the corresponding language. • If the lang element is zh-TW, display the Taiwanese era as its full name in the corresponding language. • For all other lang element values, display the Japanese Emperor Era era as its full name in ja-JP. Multiple instances of the picture item create repeated content as specified by the “G” picture item description.
M	Formats the month as a number without a leading zero for single-digit months. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “MMMM” in each group. Format each group using the description outlined for the “MMMM” picture item. 2. Repeat step 1 for groups of “MMM”, “MM”, and “M” using the respective description for each picture item group.
MM	Formats the month as a number with a leading zero for single-digit months. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb or bbbb picture item (to the Thai Buddhist Era calendar).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Multiple instances of the picture item create repeated content as specified by the “M” picture item description.
MMM	Formats the month in its abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ววววว, ววว, วว, and ว picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “M” picture item description.
MMMM	Formats the month as its full name according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, ปป, ดดดด, ดดด, ดด, ด, ววววว, ววว, วว, and ว picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “M” picture item description.
n	Formats the Japanese Emperor Era era with no leading zero for single-digit years. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “nn” in each group. Format each group using the description outlined for the “nn” picture item. 2. Repeat step 1 for groups of “n” using the description for that picture item.
nn	Formats the Japanese Emperor Era era with leading zero for single-digit years. Multiple instances of the picture item create repeated content as specified by the “n” picture item description.
O	Formats the month as a number without a leading zero for single-digit months in Japanese numerals. Multiple instances of the picture item create repeated content.
w	Formats the day of the week in an abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ja-JP or ko-KR, display in the corresponding language. • For all other lang element values, display picture item as literal text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Multiple instances of the picture item create repeated content.
W	Formats the day of the week in an abbreviated form according to the language specified by the lang element (§17.3.2.20) on the run containing the field instructions: • If the lang element is ja-JP, ko-KR, zh-TW, zh-CN, zh-HK, zh-SG, or zh-MO, display in the corresponding language. • For all other lang element values, display the picture item as literal text. Multiple instances of the picture item create repeated content.
y	Formats the year as a 2-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain "yyyy" in each group. Format each group using the description outlined for the "yyyy" picture item. 2. Repeat step 1 for groups of "yy" and "y" using the respective description for each picture item group.
yy	Formats the year as a 2-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches. Multiple instances of the picture item create repeated content as specified by the "y" picture item description.
yyyy	Formats the year as a 4-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches. Multiple instances of the picture item create repeated content as specified by the "y" picture item description.
Y	Formats the year as a 2-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, and ปป picture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the following:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	1. Working from the text direction of the run, create as many groups as possible that contain “YYYY” in each group. Format each group using the description outlined for the “YYYY” picture item. 2. Repeat step 1 for groups of “YY” and “Y” using the respective description for each picture item group.
YY	Formats the year as a 2-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, and ปปpicture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “Y” picture item description.
YYYY	Formats the year as a 4-digit number. Defaults to the Gregorian calendar [ISO 8601], but also changes in the presence of the \s and \h switches, and the bb, bbbb, ปปปป, and ปปpicture item (to the Thai Buddhist Era calendar). Multiple instances of the picture item create repeated content as specified by the “Y” picture item description.
๖	Formats the day of the month as a number without a leading zero for single-digit days in Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “๖๖” in each group. Format each group using the description outlined for the “๖๖” picture item. 2. Repeat step 1 for groups of “๖๖”, “๖๖” and “๖” using the respective description for each picture item group.
๖๖	Formats the day of the month as a two-digit number (with a leading zero for single-digit days) in Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “๖” picture item description.
๖๖๖	Formats the Thai Buddhist Era day of the week in its abbreviated form in Thai. This picture item can modify the behaviour of other picture items.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	Multiple instances of the picture item create repeated content as specified by the “ว” picture item description.
วววว	Formats the Thai Buddhist Era day of the week as its full name in Thai. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “ว” picture item description.
ด	Formats the Thai Buddhist Era month as a number without a leading zero for single-digit months in Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the following: 1. Working from the text direction of the run, create as many groups as possible that contain “ดดดด” in each group. Format each group using the description outlined for the “ดดดด” picture item. 2. Repeat step 1 for groups of “ดดด”, “ดด” and “ด” using the respective description for each picture item group.
ดด	Formats the Thai Buddhist Era month as a two-digit number (with a leading zero for single-digit months) in Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “ด” picture item description.
ดดด	Formats the Thai Buddhist Era month in its abbreviated form. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “ด” picture item description.
ดดดด	Formats the Thai Buddhist Era month as its full name. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “ด” picture item description.
ปป	Formats the Gregorian year as a 2-digit number using Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Date Formatting Picture Items	
	the following: 1. Working from the text direction of the run, create as many groups as possible that contain “ปปปป” in each group. Format each group using the description outlined for the “ปปปป” picture item. 2. Repeat step 1 for groups of “ปป” using the description for that picture item.
ปปปป	Formats the Gregorian year [ISO 8601] as a 4-digit number using Thai numerals. This picture item can modify the behaviour of other picture items. Multiple instances of the picture item create repeated content as specified by the “ปป” picture item description.

Miscellaneous Formatting Picture Items	
Picture Item	Description
`Other character`	Includes the specified character in the result at that position. [*Note*: Commonly used characters are colon (:), hyphen (-), asterisk (*), slash (/), and space. *end note*]
`'`*text*`'`	Includes *text* in the result.
`` `*numbered-item*` ``	Includes, in Arabic numerals, the number of the preceding item numbered as a caption or resulting from a SEQ field (§17.16.5.56). *numbered-item* shall be the same name as *identifier* in that SEQ field.

[*Example*: When updated in a US-English context on the date and time shown below, the following fields produced these results:

`DATE \@ "M/d/yyyy"`	`1/3/2006`
`DATE \@ "dddd, MMMM dd, yyyy"`	`Tuesday, January 03, 2006`
`DATE \@ "MMMM d, yyyy"`	`January 3, 2006`
`DATE \@ "M/d/yy"`	`1/3/06`
`DATE \@ "yyyy-MM-dd"`	`2006-01-03`
`DATE \@ "d-MMM-yy"`	`3-Jan-06`
`DATE \@ "M.d.yyyy"`	`1.3.2006`
`DATE \@ "MMM. d, yy"`	`Jan. 3, 06`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

DATE \@ "d MMMM yyyy"	3 January 2006
DATE \@ "MMMM yy"	January 06
DATE \@ "MMM-yy"	Jan-06
DATE \@ "M/d/yyyy h:mm am/pm"	1/3/2006 5:28 PM
DATE \@ "M/d/yyyy h:mm:ss am/pm"	1/3/2006 5:28:34 PM
DATE \@ "h:mm am/pm"	5:28 PM
DATE \@ "h:mm:ss am/pm"	5:28:34 PM
DATE \@ "HH:mm"	17:28
DATE \@ "'Today is 'HH:mm:ss"	Today is 17:28:34

end example]

17.16.4.2 Numeric formatting

numeric-formatting-switch=
\# *switch-argument* ;

A *numeric-formatting-switch* specifies the format of a numeric result. If the result of a field is not a number, this switch has no effect.

If no *numeric-formatting-switch* is present, a numeric result is formatted without leading spaces or trailing fractional zeros. If the result is negative, a leading minus sign is present. If the result is a whole number, no radix point is present.

A numeric *switch-argument* is made up of a series of *picture items*.

Numeric Formatting Picture Items	
Picture Item	Description
0	Specifies the requisite numeric positions to display in the result. If the result does not include a digit in that position, 0 is displayed. [*Example*: In a US-English context, =4+5 \# 00.00 displays "09.00". *end example*]
#	Specifies the requisite numeric positions to display in the result. If the result does not include a digit in that position, a space is displayed. Extra fractional digits are rounded off. [*Example*: =9+6 \# $### displays "$ 15". *end example*]
x	Drops digits to the left of the x placeholder. If the placeholder is to the right of the decimal point, the result is rounded to that place. [*Example*: In a US-English context, =111053+111439 \# x## displays "492", =1/8 \# 0.00x displays "0.125", and =3/4 \# .x displays ".8". *end example*]
.	Indicates the radix-point position. [*Example*: In a US-English context, =95.4 \# $###.00 displays "$ 95.40. *end example*] The radix-point

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Numeric Formatting Picture Items	
	character displayed is locale-specific.
,	Separates groups of three digits. [*Example*: In a US-English context, `=2456800 \# $#,###,###` displays "`2,456,800`". *end example*] The separator character displayed is locale-specific.
-	Prepends a minus sign to a negative result, or prepends a space if the result is positive or 0. [*Example*: `=80-90 \# -##` displays "`-10`", while `=90-80 \# -##` displays "` 80`". *end example*]
+	Prepends a plus sign to a positive result, a minus sign to a negative result, or a space if the result is 0. [*Example*: `=90-80 \# +##` displays "`+10`", and `=80-90 \# +##` displays "`-10`". *end example*]
`Other character`	Includes the specified character in the result at that position. [*Example*: `=33 \# ##%` displays "`33%`". *end example*]
'*text*'	Includes text in the result. [*Example*: In a US-English context, if Price is a bookmark for 26.5, `=Price*15% \# "##0.00 'is the sales tax'"` displays "`$  3.98 is the sales tax`". *end example*]
`` `numbered-item` ``	Includes, in Arabic numerals, the number of the preceding item numbered as a caption or resulting from a SEQ field (§17.16.5.56). *numbered-item* shall be the same name as *identifier* in that SEQ field. [*Example*: `=SUM(A1:D4) \# "##0.00 'is the total of Table' `table`"` displays "`456.34 is the total of Table 2`". *end example*]
positive-result ; *negative-result*	Specifies different sets of picture items for positive and negative results. A zero value uses the positive picture. [*Example*: `=Sales95 \# $#,##0.00;-$#,##0.00` displays that bookmark's positive values using `$#,##0.00`, and it's negative values using `-$#,##0.00`. *end example*]
positive-result ; *negative-result* ; *zero-result*	Specifies different sets of picture items for positive, negative, and zero results. [*Example*: `=Sales95 \# $#,##0.00;-$#,##0.00;$0` displays that bookmark's positive values using `$#,##0.00`, it's negative values using `-$#,##0.00`, and its zero values using `$0`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.4.3 General formatting

general-formatting-switch=
* *switch-argument* ;

A *general-formatting-switch* specifies a variety of formats for a numeric or text result. If the result type of a field does not correspond to the format specified, this switch has no effect.

A *switch-argument* is made up of a series of *picture items*.

17.16.4.3.1 General formatting - Numeric Values

The following *switch-argument*s apply to fields whose field result is a numeric value. If the result type of the field is not numeric, then these switches have no effect. If the field result varies based on the language of the field instructions, those variations are noted inline:

©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
Switch Argument	Description
AIUEO	Formats a numeric result using hiragana characters in the traditional a-i-u-e-o order. [*Example*: 1 * AIUEO results in ア. *end example*] Corresponds to an ST_NumberFormat enumeration value of aiueoFullWidth.
ALPHABETIC	Formats a numeric result as one or more occurrences of an uppercase alphabetic Latin character. Value 1 results in the letter A, value 2 results in the letter B, and so on up to value 26, which results in the letter Z. For values greater than 26, 26 is repeatedly subtracted from the value until the result is 26 or less. The result value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value. [*Example*: =54 * ALPHABETIC results in "BBB" as subtracting 26 from 54 two times, producesthe value 2, which is represented by the letter B. *end example*] Corresponds to an ST_NumberFormat value of upperLetter.
alphabetic	Formats a numeric result as one or more occurrences of an lowercase alphabetic Latin character. Value 1 results in the letter a, value 2 results in the letter b, and so on up to value 26, which results in the letter z. For values greater than 26, 26 is repeatedly subtracted from the value until the result is 26 or less. The result value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value. [*Example*: =52 * alphabetic results in "zz" as subtracting 26 from 52 one time, produces the value 26, which is represented by the letter z.. *end example*] Corresponds to an ST_NumberFormat enumeration value of lowerLetter.
Arabic	Formats a numeric result using Arabic cardinal numerals. [*Example*: For page 123, PAGE * Arabic results in "١٢٣". *end example*] Corresponds to an ST_NumberFormat enumeration value of decimal.
ARABICABJAD	Formats a numeric result using ascending Abjad numerals. [*Example*: 12 * ARABICABJAD results in ك. *end example*] Corresponds to an ST_NumberFormat enumeration value of arabicAbjad.
ARABICALPHA	Formats a numeric result using characters in the Arabic alphabet. [*Example*: 12 * ARABICABJAD results in س. *end example*] Corresponds to an ST_NumberFormat enumeration value of arabicAlpha.
ArabicDash	Formats a numeric result using Arabic cardinal numerals, with a prefix of "- " and a suffix of " -". [*Example*: For page 123, PAGE * ArabicDash results in "- ١٢٣ -". *end example*] Corresponds to an ST_NumberFormat enumeration value of numberInDash.
BAHTTEXT	Formats a numeric result in the following form: • If the value is an integer, it is displayed using the . Thai counting system, with บาทถ้วน appended to the result. • If the value includes a fractional value, the fractional part is rounded to two decimal places, and the resulting value is displayed in the form integer-part-in-Thai-counting format บาท fractional-part-in-Thai-counting format สตางค์. [*Example*: 1 * BAHTTEXT results in หนึ่งบาทถ้วน. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
	Corresponds to an ST_NumberFormat enumeration value of `bahtText`.
`CardText`	Formats a numeric result as lowercase cardinal text. [*Example*: For page 123, `PAGE \* CardText` results in "`one hundred twenty-three`". *end example*] Corresponds to an ST_NumberFormat enumeration value of `cardinalText`.
`CHINESENUM1`	Formats a numeric result using ascending numbers from the appropriate counting system. [*Example*: `10 \* CHINESENUM1` results in 十. *end example*] Corresponds to an ST_NumberFormat enumeration value of `chineseCounting` (zh-CN) or `taiwaneseCounting` (zn-TW).
`CHINESENUM2`	Formats a numeric result using sequential numbers from the appropriate legal format. [*Example*: `123 \* CHINESENUM2` results in 壹佰貳拾參. *end example*] Corresponds to an ST_NumberFormat enumeration value of `chineseLegalSimplified` (zh-CN) or `ideographLegalTraditional` (zh-TW).
`CHINESENUM3`	Formats a numeric result using sequential numbers from the appropriate counting thousand system. [*Example*: `10 \* CHINESENUM3` results in 一百二十三. *end example*] Corresponds to an ST_NumberFormat enumeration value of `chineseCountingThousand` (zh-CN) or `taiwaneseCountingThousand` (zh-TW).
`CHOSUNG`	Formats a numeric result using sequential numbers from the Korean Chosung format. [*Example*: `1 \* CHOSUNG` results in ㄱ. *end example*] Corresponds to an ST_NumberFormat enumeration value of `chosung`.
`CIRCLENUM`	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character for numbers in the range 1–20. For non-negative numbers outside this range, formats them as with `ARABIC`. [*Example*: `12 \* CIRCLENUM` results in ⑫. *end example*] Corresponds to an ST_NumberFormat enumeration value of `decimalEnclosedCircle`.
`DBCHAR`	Formats a numeric result using double-byte Arabic numbering. [*Example*: `123 \* DBCHAR` results in １２３. *end example*] Corresponds to an ST_NumberFormat enumeration value of `decimalFullWidth`.
`DBNUM1`	Formats a numeric result using sequential digital ideographs, using the appropriate character. [*Example*: `12 \* DBNUM1` results in 一二. *end example*] Corresponds to an ST_NumberFormat enumeration value of `ideographDigital` (ja-JP) or `koreanDigital` (ko-KR).
`DBNUM2`	Formats a numeric result using sequential numbers from the appropriatecounting system. [*Example*: `12 \* DBNUM2` results in 十二. *end example*] Corresponds to an ST_NumberFormat enumeration value of `japaneseCounting` (ja-JP) or `koreanCounting` (ko-KR).
`DBNUM3`	Formats a numeric result using sequential numbers from the appropriate legal counting system. [*Example*: `12 \* DBNUM3` results in 壹拾弐. *end example*] Corresponds to an ST_NumberFormat enumeration value of `japaneseLegal`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
	(ja-JP) or koreanLegal (ko-KR).
DBNUM4	Formats a numeric result using sequential numbers from the appropriate digital counting system. [*Example*: 12 * DBNUM4 results in 一二. *end example*] Corresponds to an ST_NumberFormat enumeration value of japaneseDigitalTenThousand (ja-JP) or koreanDigital2 (ko-KR) or taiwaneseDigital (zh-TW).
DollarText	Formats a numeric result in the following form: *integer-part-as-cardinal-text* and *nn*/100 The fractional part is rounded to two decimal places, *nn*, and is formatted using Arabic cardinal numerals. [*Example*: =1234.567 * DollarText results in "one thousand two hundred thirty-four and 57/100". *end example*] Corresponds to an ST_NumberFormat enumeration value of dollarText.
GANADA	Formats a numeric result using sequential numbers from the Korean Ganada format. [*Example*: 12 * GANADA results in 다. *end example*] Corresponds to an ST_NumberFormat enumeration value of ganada.
GB1	Formats a numeric result using decimal numbering followed by a period, using the enclosed alphanumeric glyph character. [*Example*: 12 * GB1 results in 12.. *end example*] Corresponds to an ST_NumberFormat enumeration value of decimalEnclosedFullstop.
GB2	Formats a numeric result using decimal numbering enclosed in parenthesis, using the enclosed alphanumeric glyph character. [*Example*: 12 * GB2 results in ⑿. *end example*] Corresponds to an ST_NumberFormat enumeration value of decimalEnclosedParen.
GB3	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers can be replaced with non-enclosed equivalents. [*Example*: 12 * GB3 results in 12. *end example*] Corresponds to an ST_NumberFormat enumeration value of decimalEnclosedCircleChinese.
GB4	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers can be replaced with non-enclosed equivalents. [*Example*: 12 * GB4 results in 12. *end example*] Corresponds to an ST_NumberFormat enumeration value of ideographEnclosedCircle.
HEBREW1	Formats a numeric result using Hebrew numerals. [*Example*: 123 * HEBREW1 results in קכג. *end example*] Corresponds to an ST_NumberFormat enumeration value of hebrew1.
HEBREW2	Formats a numeric result using the Hebrew alphabet. [*Example*: 123 * HEBREW2 results in תתתתתמ. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
	Corresponds to an ST_NumberFormat enumeration value of hebrew2.
Hex	Formats the numeric result using uppercase hexadecimal digits. [*Example*: For page 355, PAGE * Hex results in "FF". *end example*] Corresponds to an ST_NumberFormat enumeration value of hex.
HINDIARABIC	Formats a numeric result using Hindi numbers. [*Example*: 123 * HINDIARABIC results in १२३. *end example*] Corresponds to an ST_NumberFormat enumeration value of hindiNumbers.
HINDICARDTEXT	Formats a numeric result using sequential numbers from the Hindi counting system. [*Example*: 123 * HINDICARDTEXT results in एक सौ तेईस. *end example*] Corresponds to an ST_NumberFormat enumeration value of hindiCounting.
HINDILETTER1	Formats a numeric result using Hindi vowels. [*Example*: 123 * HINDILETTER1 results in ठठठठ. *end example*] Corresponds to an ST_NumberFormat enumeration value of hindiVowels.
HINDILETTER2	Formats a numeric result using Hindi consonants. [*Example*: 123 * HINDILETTER2 results in ओओओओओओओ. *end example*] Corresponds to an ST_NumberFormat enumeration value of hindiConsonants.
IROHA	Formats a numeric result using the Japanese iroha. [*Example*: 12 * IROHA results in 才. *end example*] Corresponds to an ST_NumberFormat enumeration value of irohaFullWidth.
KANJINUM1	Formats a numeric result using a Japanese style using the appropriate counting system. [*Example*: 12 * KANJINUM1 results in 一二. *end example*] Corresponds to an ST_NumberFormat enumeration value of koreanDigital (ko-KR), ideographDigital (ja-JP), chineseCounting (zh-CN), or taiwaneseCounting (zh-TW).
KANJINUM2	Formats a numeric result using the appropriatecounting system. [*Example*: 12 * KANJINUM2 results in 十二. *end example*] Corresponds to an ST_NumberFormat enumeration value of koreanCounting (ko-KR), chineseCountingThousand (ja-JP), chineseLegalSimplified (zh-CN), or ideographLegalTraditional (zh-TW).
KANJINUM3	Formats a numeric result using the appropriatecounting system. [*Example*: 12 * KANJINUM3 results in 壱拾弐. *end example*] Corresponds to an ST_NumberFormat enumeration value of koreanLegal (ko-KR) or japaneseLegal (ja-JP) or chineseCountingThousand (zh-CN) or taiwaneseCountingThousand (zh-TW).
Ordinal	Formats a numeric result using lowercase ordinal Arabic numerals. [*Example*: =32 * Ordinal results in "32nd". *end example*] Corresponds to an ST_NumberFormat enumeration value of ordinal.
OrdText	Formats a numeric result as lowercase ordinal text. Apart from being used to round off the whole number part, the fractional part is not used. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
	=1234.567 * OrdText results in "one thousand two hundred thirty-fifth". *end example*] Corresponds to an ST_NumberFormat enumeration value of ordinalText.
Roman	Formats a numeric result using uppercase Roman numerals. [*Example*: For page 123, PAGE * Roman results in "CXXIII". *end example*] Corresponds to an ST_NumberFormat enumeration value of upperRoman.
roman	Formats a numeric result using lowercase Roman numerals. [*Example*: For page 123, PAGE * roman results in "cxxiii". *end example*] Corresponds to an ST_NumberFormat enumeration value of lowerRoman.
SBCHAR	Formats a numeric result using single-byte Arabic numbering. [*Example*: 123 * SBCHAR results in 123. *end example*] Corresponds to an ST_NumberFormat enumeration value of decimalHalfWidth.
THAIARABIC	Formats a numeric result using Thai numbers. [*Example*: 123 * THAIARABIC results in ๑๒๓. *end example*] Corresponds to an ST_NumberFormat enumeration value of thaiNumbers.
THAICARDTEXT	Formats a numeric result using sequential numbers from the Thai counting system. [*Example*: 123 * THAICARDTEXT results in หนึ่งร้อยยี่สิบสาม. *end example*] Corresponds to an ST_NumberFormat enumeration value of thaiCounting.
THAILETTER	Formats a numeric result using Thai letters. [*Example*: 30 * THAILETTER results in ฮฮฮ. *end example*] Corresponds to an ST_NumberFormat enumeration value of thaiLetters.
VIETCARDTEXT	Formats a numeric result using Vietnamese numerals. [*Example*: 12 * VIETCARDTEXT results in mười hai. *end example*] Corresponds to an ST_NumberFormat enumeration value of vietnameseCounting.
ZODIAC1	Formats a numeric result using sequential numerical traditional ideographs. [*Example*: 1 * ZODIAC1 results in 甲. *end example*] Corresponds to an ST_NumberFormat enumeration value of ideographTraditional.
ZODIAC2	Formats a numeric result using sequential zodiac ideographs. [*Example*: 1 * ZODIAC2 results in 子. *end example*] Corresponds to an ST_NumberFormat enumeration value of ideographZodiac.
ZODIAC3	Formats a numeric result using sequential traditional zodiac ideographs. [*Example*: 1 * ZODIAC3 results in 甲子. *end example*] Corresponds to an ST_NumberFormat enumeration value of ideographZodiacTraditional.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.4.3.2 General formatting - String Values

The following *switch-arguments* apply to fields whose field result is a string value:

 ©ISO/IEC 2016 – All rights reserved

General Formatting Switch Arguments	
Switch Argument	Description
Caps	Capitalizes the first letter of each word. [*Example*: USERNAME "mary smith" * Caps results in "Mary Smith", whereas USERNAME "marysmith" * Caps results in "Marysmith". *end example*]
FirstCap	Capitalizes the first letter of the first word. [*Example*: USERNAME "mary smith" * FirstCap results in "Mary smith". *end example*]
Lower	All letters are lowercase. [*Example*: USERNAME "Mary Smith" * Lower results in "mary smith". *end example*]
Upper	All letters are uppercase. [*Example*: USERNAME "Mary Smith" * Upper results in "MARY SMITH". *end example*]

17.16.4.3.3 General formatting - Field Result Formatting

The following *switch-arguments* apply to any field result, and provide directions to applications regarding the formatting which should be applied to a field result after a field update has been performed. As discussed in §17.16, as to when a field update is performed is outside the scope of ISO/IEC 29500.

The general formatting switch argument CHARFORMAT is an instruction that specifies formatting intended for application to the field result after a field update. If this switch is present, the formatting of the run containing the first instrText element after the fldChar element with a fldCharType attribute value of begin is applied to all runs in the field result whenever a new field result is generated. [*Example*: In a US-English context, on January 4, 2006, the field DATE * CHARFORMAT results in "1/4/2006". However, if the first run within the field instructions (containing the D in DATE) is bold, the field **D**ATE * CHARFORMAT results in "**1/4/2006**". If that D is italic, the field *D*ATE * CHARFORMAT results in "*1/4/2006*". If that D is bold, underlined, and red, the field **D**ATE * CHARFORMAT results in "**1/4/2006**". *end example*]

The XML for the bold, underlined, red case is as follows:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> </w:instrText>
</w:r>
<w:r …>
  <w:rPr>
    <w:b/>
    <w:color w:val="ED1C24"/>
    <w:u w:val="single"/>
  </w:rPr>
  <w:instrText>D</w:instrText>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:instrText xml:space="preserve">ATE </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:t>1/4/2006</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

Then use of the CHARFORMAT switch would cause the new field result to reuse the formatting on the first run in the field instructions, like this:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> </w:instrText>
</w:r>
<w:r …>
  <w:rPr>
    <w:b/>
    <w:color w:val="ED1C24"/>
    <w:u w:val="single"/>
  </w:rPr>
  <w:instrText>D</w:instrText>
</w:r>
<w:r>
  <w:instrText xml:space="preserve">ATE /* CHARFORMAT</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:rPr>
    <w:b/>
    <w:color w:val="ED1C24"/>
    <w:u w:val="single"/>
  </w:rPr>
  <w:t>1/4/2006</w:t>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

end example]

If a format specified directly in the first run of a field's *field-type* name conflicts with a general formatting switch, the general formatting switch is ignored. [*Example*: If the first run is set in small caps and the switch `\* Lower` is also used, that switch is ignored. *end example*]

The general formatting switch argument `MERGEFORMAT` is used to specify an instruction regarding the formatting which is applied to the field result after a field update. If this switch is present, it specifies that applications that perform field updates should preserve the formatting of the existing field result when a new field result is generated, by applying the following logic:

- Delete all text from the current field result (leaving the paragraph and run structure intact).
- Insert the new field result text into the existing run/paragraph structure.
  - If the new text does not fill the existing structure, delete the superfluous runs/paragraphs.
  - If the new text overflows the existing structure, add additional runs/paragraphs as needed.

[*Example*: Consider the following field:

```
TIME \@ "HH:mm:ss" \* MERGEFORMAT
```

When it is updated, the result might be `12:22:27` for example. If the seconds part of the displayed field result was underlined by the use of direct formatting within the original WordprocessingML, as in 12:22:27, when that field is next updated, the seconds underlining is preserved.

If the original XML generated for this field is:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText xml:space="preserve"> TIME \@ "HH:mm:ss" \* MERGEFORMAT
</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:t>17:02:</w:t>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:r …>
  <w:rPr>
    <w:u w:val="single"/>
  </w:rPr>
  <w:t>32</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

then use of the MERGEFORMAT switch would cause the new field result to reuse that structure, like this:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText>TIME \@ "HH:mm:ss" \* MERGEFORMAT</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r …>
  <w:t>12:22:</w:t>
</w:r>
<w:r …>
  <w:rPr>
    <w:u w:val="single"/>
  </w:rPr>
  <w:t>27</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

and omitting it would direct an application that it could replace it with a single new run, like this:

```
<w:r>
  <w:fldChar w:fldCharType="begin"/>
</w:r>
<w:r>
  <w:instrText>TIME \@ "HH:mm:ss"</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:t>12:22:27</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

end example]

17.16.5 Field definitions

Each of the subclauses below this subclause describes a separate field, and each description contains a section marked **Syntax**. That section contains pieces of the field grammar as they pertain to that specific field. These pieces are presented in a slightly simpler form to aid in the understanding of the description. In those sections, the left-square-bracket and right-square-bracket designate the start and end of an option, as used in ISO/IEC 14977. However, the field-name, the open-parenthesis and the close-parenthesis designate actual literal text, as does each comma. [*Note*: Therefore, in a strict presentation according to ISO/IEC 14977, each field-name, open-parenthesis and close-parenthesis would appear with double-quotes surrounding each instance. *end note*]

The set of fields is divided into the following functional categories:

Category	Description	Fields
Date and Time	Inserts the current date and/or time, or date and/or time of some kind of event.	CREATEDATE (§17.16.5.11), DATE (§17.16.5.13), EDITTIME (§17.16.5.16), PRINTDATE (§17.16.5.47), SAVEDATE (§17.16.5.53), TIME (§17.16.5.65)
Document Automation	Provides functionality for automated document processing.	COMPARE (§17.16.5.10), DOCVARIABLE (§17.16.5.15), GOTOBUTTON (§17.16.5.23), IF (§17.16.5.26), MACROBUTTON (§17.16.5.34), PRINT (§17.16.5.46)
Document Information	Inserts or stores information about the document.	AUTHOR (§17.16.5.4), COMMENTS (§17.16.5.9), DOCPROPERTY (§17.16.5.14), FILENAME (§17.16.5.17), FILESIZE (§17.16.5.18), KEYWORDS (§17.16.5.30), LASTSAVEDBY (§17.16.5.31), NUMCHARS (§17.16.5.41), NUMPAGES (§17.16.5.42), NUMWORDS (§17.16.5.43), SUBJECT (§17.16.5.60), TEMPLATE (§17.16.5.64), TITLE (§17.16.5.66)
Equations and Formulas	Defines formulas and calculates results; inserts symbols.	= *formula* (§17.16.3), ADVANCE (§17.16.5.2), SYMBOL (§17.16.5.61)
Form Fields	Defines fields that support insertion of data through	FORMCHECKBOX (§17.16.5.20), FORMDROPDOWN (§17.16.5.21),

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Category	Description	Fields
	form controls.	FORMTEXT (§17.16.5.22)
Index and Tables	Defines entries for, and builds, a table of contents, table of figures, or table of authorities.	INDEX (§17.16.5.29), RD (§17.16.5.50), TA (§17.16.5.62), TC (§17.16.5.63), TOA (§17.16.5.67), TOC (§17.16.5.68), XE (§17.16.5.72)
Links and References	Inserts information from another place in the same document or from a different document or file.	AUTOTEXT (§17.16.5.5), AUTOTEXTLIST (§17.16.5.6), BIBLIOGRAPHY (§17.16.5.7), CITATION (§17.16.5.8) HYPERLINK (§17.16.5.25), INCLUDEPICTURE (§17.16.5.27), INCLUDETEXT (§17.16.5.28), LINK (§17.16.5.32), NOTEREF (§17.16.5.40), PAGEREF (§17.16.5.45), QUOTE (§17.16.5.49), REF (§17.16.5.51), STYLEREF (§17.16.5.59)
Mail Merge	Defines information that is to be used in a mail merge. A *mail merge* is a process by which a data set (e.g., of names and addresses) is combined with a WordprocessingML document to produce a customized document for each record in said data set. In other words, a *mail merge* is an operation by which an application replaces certain fields with the data in each record from a corresponding data source (see §17.14 for additional information).	ADDRESSBLOCK (§17.16.5.1), ASK (§17.16.5.3), COMPARE (§17.16.5.10), DATABASE (§17.16.5.12), FILLIN (§17.16.5.19), GREETINGLINE (§17.16.5.24), IF (§17.16.5.26), MERGEFIELD (§17.16.5.35), MERGEREC (§17.16.5.36), MERGESEQ (§17.16.5.37), NEXT (§17.16.5.38), NEXTIF (§17.16.5.39), SET (§17.16.5.57), SKIPIF (§17.16.5.58)
Numbering	Specifies numbering for document items such as sections and pages; also bar codes.	LISTNUM (§17.16.5.33), PAGE (§17.16.5.44), REVNUM (§17.16.5.52), SECTION (§17.16.5.54), SECTIONPAGES (§17.16.5.55), SEQ (§17.16.5.56)
User Information	Referencesthe name, initials, or address of a user account under which the document is manipulated. [*Note*: These fields can be used in documents to allow applications to perform implementation-defined updates under a particular	USERADDRESS (§17.16.5.69), USERINITIALS (§17.16.5.70), USERNAME (§17.16.5.71)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Category	Description	Fields
	user's context (if such a context exists); for example, to add the address of the current user to a generic form letter. *end note*]	

17.16.5.1 ADDRESSBLOCK

Syntax:

ADDRESSBLOCK [*switches*]

Description: Represents an address block. An *address block* is a block of text specifying information appropriate for a postal mailing address, in the order required by the destination country. [*Example*: An address block for the Canadian postal system might consist of:

```
<<RECIPIENT NAME>>
<<POSTAL ADDRESS>>
<<CITY>>, <<PROVINCE>>, Canada
<<POSTAL CODE>>
```

end example]

Field Value: The address block.

Switches: Zero or more of the following *field-specific-switches*.

\c *field-argument*	*text* in this switch's *field-argument* specifies whether to include the name of the country/region: a value of 0 causes the country/region to be omitted; a value of 1 causes it to be included (regardless of any value given for \e), and a value of 2 causes country/region to be included, but only if it is different from any and all specifiedvalues for \e.
\d	Specifies that the address is to be formatted according to the country/region of the recipient as defined by POST*CODE (Universal Postal Union 2006). If this switch is not used, then addresses are formatted according to the ordering defined by the \f switch. If that switch is also omitted, addresses are formatted according to some implementation-specific preference.
\e *field-argument*	*text* in this switch's *field-argument* specifies acountry/region to exclude from the address block. [*Note*: This is useful when your mailing contains a mix of domestic and international recipients. *end note*] To exclude more than one country or region, use a \e switch for each one, as multiple \e switches are permitted.
\f *field-argument*	*text* in this switch's *field-argument* specifies the name and address format by providing a template of merge-field placeholders in the following format: "<<_" + MAPPED NAME + "_>>" (< = U+003C; > = U+003E; _ = U+005F), where MAPPED NAME corresponds to a value specified by the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr>
<td></td>
<td>mappedName element (§17.14.23).

The following possible values of MAPPED NAME and corresponding mapped merge field names are reserved:

<table>
<tr><th>MAPPED NAME</th><th>Merge Field Name</th></tr>
<tr><td>TITLE0</td><td>Title</td></tr>
<tr><td>FIRST0</td><td>First Name</td></tr>
<tr><td>LAST0</td><td>Last Name</td></tr>
<tr><td>SUFFIX0</td><td>Suffix</td></tr>
<tr><td>COMPANY</td><td>Company</td></tr>
<tr><td>STREET1</td><td>Address 1</td></tr>
<tr><td>STREET2</td><td>Address 2</td></tr>
<tr><td>CITY</td><td>City</td></tr>
<tr><td>STATE</td><td>State</td></tr>
<tr><td>POSTAL</td><td>Postal Code</td></tr>
<tr><td>COUNTRY</td><td>Country or Region</td></tr>
</table>

[Example: Using these values, a U.S. postal address might be specified as follows:

<pre>\f "<<_TITLE0_>><<_FIRST0_>><<< _LAST0_>><<< _SUFFIX0_>>
<<_COMPANY_>>
<<_STREET1_>>
<<_STREET2_>>
<<_CITY_>><<<, _STATE_>><<< _POSTAL_>><<<_COUNTRY_>>"</pre>

end example]</td>
</tr>
<tr>
<td>\l field-argument</td>
<td>text in this switch's field-argument specifies the language ID used to format the address. The default is to use the language ID of the first character of the document. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)</td>
</tr>
</table>

17.16.5.2 ADVANCE

Syntax:

ADVANCE [*switches*]

Description: When text within a document is rendered, typically, each character is displayed immediately following the previous piece of content (another character, an image, and so on), according to the lexical order of the underlying WordprocessingML. The presence of this field moves the starting point at which the text that lexically follows the field is displayed to the right or left, up or down, or to a specific horizontal or vertical position. It does not affect the display of other run content (e.g., images). The switches used by this field can

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

cause text to overlap when it is displayed. If the text is displayed within the constraints of a fixed page, repositioned text is not display if it is moved beyond the print margins of the current page (even if the target location would logically appear on a previous/following page).

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

`\d` *field-argument*	Moves the text that follows the field down by the integral number of points specified by *text* in this switch's *field-argument*.
`\l` *field-argument*	Moves the text that follows the field left by the integral number of points specified by *text* in this switch's *field-argument*.
`\r` *field-argument*	Moves the text that follows the field right by the integral number of points specified by *text* in this switch's *field-argument*.
`\u` *field-argument*	Moves the text that follows the field up by the integral number of points specified by *text* in this switch's *field-argument*.
`\x` *field-argument*	Moves the text that follows the field the integral number of points specified by *text* in this switch's *field-argument* from the left edge of the column, frame, or text box.
`\y` *field-argument*	Moves the text that follows the field the integral number of points specified by *text* in this switch's *field-argument*. This shift is the vertical position relative to the page. The entire line of text that contains the field is moved. This switch is ignored if it specifies a location outside the page margins or if the switch is used inside any of the following: table, text box, footnote, endnote, annotation, header, or footer.

[*Example*: When the following WordprocessingML is displayed:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:r>
  <w:t>XX</w:t>
</w:r>
<w:fldSimple w:instrText="ADVANCE \u 6"/>
<w:r>
  <w:t>XX</w:t>
</w:r>
<w:fldSimple w:instrText="ADVANCE \d 12"/>
<w:r>
  <w:t>XX</w:t>
</w:r>
<w:fldSimple w:instrText="ADVANCE \l 20"/>
<w:r>
  <w:t>+</w:t>
</w:r>
<w:fldSimple w:instrText="ADVANCE \x 150"/>
<w:r>
  <w:t>ZZ</w:t>
</w:r>
```

The results are:

XX XX
XX ZZ

end example]

17.16.5.3 ASK

Syntax:

ASK *field-argument-1 field-argument-2* [*switches*]

field-argument-1:
field-argument

field-argument-2:
field-argument

Description: Prompts the user to enter information and assigns the bookmark designated by *field-argument-1* to represent the user's response. *text* in *field-argument-2* specifies the prompt text, which is displayed in a dialog box. The prompt is displayed each time the ASK field is updated. A response remains assigned to the bookmark until a new response is entered.Represents information needed to elicit and store a response (where a response is any string of characters, including the empty string). When a field update is performed, the text specified by *field-argument-2* is intended for use when requesting the response, and the resulting response is stored within the bookmark (§17.18.29) whose name attribute matches the value specified by *field-argument-1*.

Field Value: None.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Switches: Zero or more of the following *field-specific-switches*.

\d *field-argument*	The *text* in this switch's *field-argument* specifies a default response if one is not entered. If no default response is specified, the *field* result remains unchanged. To specify a blank entry as the default, *field-argument* shall be "".
\o	When used in a mail merge main document, this causes the display of the prompt once instead of each time a new data record is merged. The same response is inserted in each resulting merged document.

[*Example*: When the following fields are updated and "John" is entered as the response,

```
ASK AskResponse "What is your first name?"
Hello REF AskResponse.
```

the result is

```
Hello John.
```

end example]

17.16.5.4 AUTHOR

Syntax:

AUTHOR [*field-argument*] [*switches*]

Description: Retrieves, and optionally sets, the document author's name, as recorded in the Creator element of the Core File Properties part or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a *field-argument* shall change Creator to *text*.

Field Value: The document author's name.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

[*Example*: Consider the case in which the Creator element is as follows:

```
<Creator>William Jones</Creator>
```

and the following field is updated:

```
AUTHOR
```

The result is:

```
William Jones
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Updating the following field:

```
AUTHOR "Tony Caruso"
```

causes the Creator element to take on the specified value. *end example*]

17.16.5.5 AUTOTEXT

Syntax:

AUTOTEXT *field-argument*

Description: References theAutoText entry (§**17.12.5**) that meets the following conditions:

- The val attribute of the gallery element (§**17.12.9**) has a value of autoTxt.
- The val attribute of the name element (§**17.12.13**) has the value specified by *text* in *field-argument*.

When a field update is performed, the field result is the WordprocessingML contents of the AutoText entry.

Field Value: The WordprocessingML content of the specified AutoText entry.

Switches: None.

[*Example*: Assuming the following entries are defined with values of current page number, salutation, and a notice:

```
AUTOTEXT "- PAGE -"
AUTOTEXT "Yours truly,"
AUTOTEXT Confidential
```

when evaluated, they might produce the following results.

```
 - 13 -
Yours truly,
CONFIDENTIAL
```

end example]

17.16.5.6 AUTOTEXTLIST

Syntax:

AUTOTEXTLIST *field-argument* [*switches*]

Description: Creates a shortcut menu based on AutoText entries in the active template. The list can vary based on the styles applied to the AutoText entries. *text* in *field-argument* is inserted into the document.

A complex field XML implementation shall be used, and the *field-argument* text shall be placed in one or more runs between the separate and end parts of the fldChar element.

Field Value: *text* in *field-argument*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Switches: Zero or more of the following *field-specific-switches*.

`\s` *field-argument*	Specifies that the list is to contain entries based on the style specified by *text* in this switch's *field-argument*. Without this switch, entries of the current paragraph style appear. If there are no entries for the current style, all entries appear. The style can be a paragraph style or a character style.
`\t` *field-argument*	*text* in this switch's *field-argument* specifies the text to show in the ScreenTip.

[*Example*: The field:

```
{ AUTOTEXTLIST "List of salutations" \s Salutation \t "Choose a salutation" }
```

causes the following to be displayed: In the document, the Salutation list; in the ScreenTip, `Choose a salutation`, and on the shortcut menu, the list of entries whose style is `Salutation`.

end example]

17.16.5.7 BIBLIOGRAPHY

Syntax:

`BIBLIOGRAPHY` [*switches*]

Description: Retrieves and displays the contents of the document's Bibliography part in the bibliographic style specified within the SelectedStyle attribute of the Sources (§22.6.2.60) element of the Bibliography part.

Field Value: The formatted bibliographic data for all sources in the current document.

Switches: Zero of more of the following *field-specific-switches*.

`\l` *field-argument*	The *text* in this switch's *field-argument* specifies the language ID that shall be used in to format the bibliographic sources in the document that do not specify a locale using the LCID (§22.6.2.39) element. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)
`\f` *field-argument*	The *text* in this switch's *field-argument* specifies the language ID that shall be used to filter the bibliographic data to only the sources in the document that use that language. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)
`\m` *field-argument*	The *text* in this switch's *field-argument* specifies that only the source with a Tag (§22.6.2.65) element value matching *field-argument* shall be displayed in the bibliography.

[*Example:* Consider a document with bibliographic data for ISO/IEC 29500:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Author: Ecma International
Title: Office Open XML Document Interchange Specification
Year: 2006
Month: October

the following field is updated:

```
BIBLIOGRAPHY /l 1033
```

The result for MLA is:

```
Ecma International. Office Open XML Document Interchange Specification. October
2006.
```

And for APA:

```
Ecma International. (2006, October). Office Open XML Document Interchange
Specification.
```

end example]

17.16.5.8 CITATION

Syntax:

CITATION *field-argument* [*switches*]

Description: Displays the contents of the Source (§22.6.2.59) element with a Tag (§22.6.2.65) element value matching *field-argument* using the bibliographic style specified within the SelectedStyle attribute of the Sources (§22.6.2.60) element of the Bibliography part.

Field Value: The comments relating to the current document.

Switches: Zero or more of the following *field-specific-switches*.

\l *field-argument*	The *text* in this switch's *field-argument* specifies the language ID which shall be used in conjunction with the specified bibliographic style to format the citation in the document. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)
\f *field-argument*	The *text* in this switch's *field-argument* specifies the prefix which shall be prepended to the citation.
\s *field-argument*	The *text* in this switch's *field-argument* specifies the suffix which shall be appended to the citation.
\p *field-argument*	The *text* in this switch's *field-argument* specifies the page number associated with the citation.
\v *field-argument*	The *text* in this switch's *field-argument* specifies the volume number associated with the citation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

\n	Specifies that the author information shall be suppressed from the citation.
\t	Specifies that the title information shall be suppressed from the citation.
\y	Specifies that the year information shall be suppressed from the citation.
\m *field-argument*	The *text* in this switch's *field-argument* specifies the Tag (§22.6.2.65) element value for another source to be included in this citation's field result.

[*Example:* Consider a case with bibliographic data for ISO/IEC 29500:

Tag: Ecma01
Author: Ecma International
Title: Office Open XML Document Interchange Specification
Year: 2006
Month: October

the following field is updated:

```
CITATION Ecma01 /l 1033
```

The result for MLA is:

```
(Ecma International)
```

And for APA:

```
(Ecma International, 2006)
```

end example]

17.16.5.9 COMMENTS

Syntax:

COMMENTS [*field-argument*] [*switches*]

Description: Retrieves, and optionally sets, the comments relating to the current document, as recorded in the Description element of the Core File Properties part or, if *field-argument* is present, the comments specified by *text* in *field-argument*. Specifying a field-argument shall change Description to *text*.

Field Value: The comments relating to the current document.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

[*Example*: Consider the case in which the Description element is as follows:

```
<Description>Once upon a time, in a land far, far away …</Description>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

and the following field is updated:

COMMENTS

The result is:

Once upon a time, in a land far, far away …

Updating the following field:

COMMENTS "I came, I saw, I was not impressed."

causes the Description element to take on the specified value. *end example*]

17.16.5.10 COMPARE

Syntax:

COMPARE *comparison* [*switches*]

Description: Compares the values designated by the two *expressions* in *comparison* using the operator designated by *comparison-operator*. [*Note*: This field can be used to create compound logical comparisons with AND and OR functions in a formula, and then by using the result of the formula in an IF field. *end note*]

[*Note*: *comparison-operator* can be any one of the six relational and equality operators specified for *operator* (§17.16.3.3). *end note*]

If *comparison-operator* is = or <>, the left-hand *expression* operand can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters. The expression shall be enclosed in quotation marks so that it is compared as a character string.

Field Value: 1 if the comparison is true, or 0 if the comparison is false.

Switches: None.

[*Example*: Consider the case in which the IF field in the following example is inserted into a mail merge main document. The COMPARE fields examine the data fields CustomerNumber and CustomerRating as each data record is merged. The OR function of the formula returns the value 1 if at least one of the data fields indicates poor credit, in which case the first text in quotation marks is printed:

```
{ IF { = OR ( { COMPARE { MERGEFIELD CustomerNumber } >= 4 },
  { COMPARE { MERGEFIELD CustomerRating } <= 9 } ) } = 1
  "Credit not acceptable" "Credit acceptable"}
```

The following COMPARE field results in the value 1 if any value in the PostalCode data field is the range 98500–98599:

```
{ COMPARE "{ MERGEFIELD PostalCode }" = "985*" }
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

17.16.5.11 CREATEDATE

Syntax:

CREATEDATE [*switches*]

Description: Retrieves the date and time at which the document was created, as recorded in the DateCreated element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-and-time-formatting-switch* used is implementation-defined.

Field Value: The date and time at which the document was created.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri Lunar or Hebrew Lunar calendar, depending on the language specified by the lang element (§17.3.2.20).
\s	Use the Saka Era calendar.

[*Example*: Consider the case in which the DateCreated element is as follows:

<DateCreated>2006-01-05T03:31:00Z</DateCreated>

and the following fields are updated in a US-English context that is UTC -5:

CREATEDATE
CREATEDATE \@ "dddd, MMMM dd, yyyy HH:mm:ss"
CREATEDATE \@ "dddd, MMMM dd, yyyy HH:mm:ss" \h
CREATEDATE \@ "dddd, MMMM dd, yyyy HH:mm:ss" \s

the results are:

```
1/4/2006 10:31:00 PM
Wednesday, January 04, 2006 22:31:00
AlArbia'a, Thoul Hijjah 04, 1426 22:31:00
Budhavara, Pausa 14, 1927 22:31:00
```

end example]

17.16.5.12 DATABASE

Syntax:

DATABASE [*switches*]

DATABASE [switches *]*

Description: Inserts the results of a database query into a WordprocessingML table. If the number of columns is 62 or more, the field inserts the results of a query in columns separated by tabs. The DATABASE field contains all

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

the information needed to connect to a database and perform an SQL query. Each time the field is updated, the database is queried again.

Field Value: The results of a database query as a WordprocessingML table.

Switches: Zero or more of the following *field-specific-switches*.

\b *field-argument*	The *text* in this switch's *field-argument* specifies which attributes of the format set by the \l switch are to be applied to the table. If the \l switch is blank, the \b switch value shall be 16 (AutoFit). *text* can have a value that is the bitwise-or of any combination of the following: 0, None 1, Borders 2, Shading 4, Font 8, Color 16, AutoFit 32, Heading Rows 64, Last Row 128, First Column 256, Last Column
\c *field-argument*	The *text* in this switch's *field-argument* specifies a connection to the data.
\d *field-argument*	The *text* in this switch's *field-argument* specifies the complete path and file name of the database. Used for all database queries except a query to an SQL database table using ODBC.
\f *field-argument*	The *text* in this switch's *field-argument* specifies the integral record number of the first data record to insert
\h	Inserts the field names from the database as column headings in the resulting table.
\l *field-argument*	The *text* in this switch's *field-argument* specifies the format that is to be applied to the result of the database query. If this switch is used and the \b switch doesn't specify the table attributes, an unformatted table is inserted.
\o *field-argument*	Inserts data at the beginning of a merge. By adding the \o switch to the database field, it only gets the data for the database field at the beginning of a merge instead of once for each record merged. This is a performance optimization and should only be used when the database field doesn't rely on record specific information to gather.
\s *field-argument*	The *text* in this switch's *field-argument* specifies a set of SQL instructions. Each quotation mark in the instructions shall be preceded by a backslash (\).
\t *field-argument*	The *text* in this switch's *field-argument* specifies the integral record number of the last data record to insert.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: The following field results from a query to a database through ODBC:

```
{ DATABASE \d "C:\\Data\\Sales93.mdb" \c "DSN=MS Access Database;
  DBQ=C:\\Data\\Sales93.mdb; FIL=RedISAM"
  \s "select * from \"Customer List\"" \f "2445" \t "2486" \l "2"
```

end example]

17.16.5.13 DATE

Syntax:

DATE [*switches*]

DATE [switches *]*

Description: The current date and time. If no calendar is specified via a *field-specific-switch* or *date-and-time-formatting-switch*,the Gregorian calendar [ISO 8601, §3.2.1]is used. As specified in §17.16.4.1, if no *date-and-time-formatting-switch* is present, a date or time result is formatted in an implementation-defined manner.

Field Value: The current date and time.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri Lunar or Hebrew Lunar calendar, depending on the language specified by the lang element (§17.3.2.20).
\l	When a field update is performed, if no *date-and-time-formatting-switch* is used, this switch is an instruction specifying that the field shall use the *date-and-time-formatting-switch* last used by the hosting application when inserting a new DATE field. If there is no last-used date format available, then the *date-and-time-formatting-switch* used is implementation-defined.
\s	Use the Saka Era calendar.

[*Example*: Consider the case in which the following fields are updated in a US-English context by an application with a default *date-and-time-formatting-switch* of `M/d/yyyy` on `2006-01-05T19:09:01`:

```
DATE
DATE \@ "dddd, MMMM dd, yyyy  HH:mm:ss"
DATE \@ "dddd, MMMM dd, yyyy  HH:mm:ss" \h
DATE \@ "dddd, MMMM dd, yyyy  HH:mm:ss" \s
```

the results are:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
1/5/2006
Thursday, January 05, 2006  19:09:01
AlKhamis, Thoul Hijjah 05, 1426  19:09:01
Bruhaspathivara, Pausa 15, 1927  19:09:01
```

end example]

[*Note*: For some *date-and-time-formatting-switch*es, the DATE and TIME (§17.16.5.65) fields can produce the same result. *end note*]

17.16.5.14 DOCPROPERTY

Syntax:

DOCPROPERTY *docprop-category* [*switches*]

Description: Retrieves the indicated document information. For some combinations of DOCPROPERTY and *docprop-category*, there is an equivalent field, in which case, the description for the combination can be obtained from that field. For those combinations not having an equivalent field, the description is shown directly. When used directly, some of the equivalent fields allow the value of the designated property to be changed. However, when the corresponding DOCPROPERTY field is used, such values shall not be changed. This is indicated in the following table by "Read-only operation."

docprop-category	Corresponding Field	Description
AUTHOR	AUTHOR (§17.16.5.4)	**Read-only operation.**
BYTES	FILESIZE (§17.16.5.18)	
CATEGORY	No equivalent	The contents of the Category element of the Core File Properties part.
CHARACTERS	NUMCHARS (§17.16.5.41)	
CHARACTERSWITHSPACES	No equivalent	**Like NUMCHARS, but includes all white space characters as well.**
COMMENTS	COMMENTS (§17.16.5.7)	**Read-only operation.**
COMPANY	No equivalent	The contents of the Company element of the Application-Defined File Properties part.
CREATETIME	CREATEDATE (§17.16.5.11)	
HYPERLINKBASE	No equivalent	The contents of the HyperlinkBase element of the Application-Defined File Properties

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

docprop-category	Corresponding Field	Description
		part.
KEYWORDS	No equivalent	The contents of the Keywords element of the Core File Properties part.
LASTPRINTED	PRINTDATE (§17.16.5.47)	
LASTSAVEDBY	LASTSAVEDBY (§17.16.5.31)	
LASTSAVEDTIME	SAVEDATE (§17.16.5.53)	
LINES	No equivalent	The contents of the Lines element of the Application-Defined File Properties part.
MANAGER	No equivalent	The contents of the Manager element of the Application-Defined File Properties part.
NAMEOFAPPLICATION	No equivalent	The contents of the Application element of the Application-Defined File Properties part.
ODMADOCID		
PAGES	NUMPAGES (§17.16.5.42)	
PARAGRAPHS	No equivalent	The contents of the Paragraphs element of the Application-Defined File Properties part.
REVISIONNUMBER	REVNUM (§17.16.5.52)	
SECURITY	No equivalent	The contents of the DocSecurity element of the Application-Defined File Properties part.
SUBJECT	SUBJECT (§17.16.5.60)	**Read-only operation.**
TEMPLATE	TEMPLATE (§17.16.5.64)	
TITLE	TITLE (§17.16.5.66)	**Read-only operation.**
TOTALEDITINGTIME	EDITTIME (§17.16.5.16)	
WORDS	No equivalent	The contents of the Words element of the Application-Defined File

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

docprop-category	Corresponding Field	Description
		Properties part.

Field Value: The indicated document information.

17.16.5.15 DOCVARIABLE

Syntax:

DOCVARIABLE *field-argument*

DOCVARIABLE field-argument

Description: Inserts the string assigned to the document variable designated by *text* in *field-argument*. Each WordprocessingML document has a collection of variables. This field is used to access and display the contents of docVar (§17.15.1.31) elements in the Document Settings part.

Field Value: The value of the specified document variable.

Switches: None.

17.16.5.16 EDITTIME

Syntax:

EDITTIME [*switches*]

Description: Retrieves the total editing time, in minutes, since the document was created, as recorded in the TotalTime element of the Application-Defined File Properties part. By default, the *numeric-formatting-switch* or *general-formatting-switch* used is implementation-defined.

Field Value: The total editing time, in minutes.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

[*Example*: Consider the case in which the TotalTime element is as follows:

<TotalTime>930</TotalTime>

and the following fields are updated in a US-English context:

EDITTIME
EDITTIME * OrdText
EDITTIME \# "#,##0"

the results are:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
930
nine hundred thirtieth
 930
```

end example]

17.16.5.17 FILENAME

Syntax:

FILENAME [*switches*]

Description: Retrieves the name of the current document from its storage location.

Field Value: The name of the current document.

Switches: One of the following *general-formatting-switch*es: * Caps, * FirstCap, * Lower, or * Upper, and zero or one of the following *field-specific-switches*.

\p	Include the full file path name.

[*Example*: Consider the case in which the following fields are updated:

```
FILENAME \* Upper
FILENAME \p
```

the results might be:

```
FIELD DEMO SUITE.DOCX
E:\Std\OOXML\Fields\Field Demo Suite.docx
http://example.com/files/myFile.docx
```

end example]

17.16.5.18 FILESIZE

Syntax:

FILESIZE [*switches*]

Description: The size of the WordprocessingML package in bytes, or the empty string if the size of the package cannot be determined when a field update is performed. [*Example*: If the package is being streamed, and its size is not yet known by the application reading that stream. *end example*] [*Note*: This information is not stored inside the document's XML. *end note*]

Field Value: The size of the current document in bytes.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch* and zero or one of the following *field-specific-switches*.

\k	Round to the nearest thousand bytes.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

\m	Round to the nearest million bytes.

[*Example*: Consider the case in the document's size is 4,660,736 bytes and the following fields are updated:

```
FILESIZE \# #,##0
FILESIZE \k
FILESIZE \m
```

the results are:

```
4,660,736
4661
5
```

end example]

17.16.5.19 FILLIN

Syntax:

FILLIN [*field-argument*] [*switch*]

Description: Prompts the user to enter text. *text* in *field-argument* contains the prompt. The prompt is displayed each time the field is updated. When a new document is created based on a template containing FILLIN fields, those fields are updated automatically.

Field Value: The user's response.

Switches: Zero or more of the following *field-specific-switches*.

\d *field-argument*	The *text* in this switch's *field-argument* specifies a default response if one is not entered. If no default response is specified, the most recent response is used. To specify a blank entry as the default, *field-argument* shall be "".
\o	When used in a mail merge main document, this causes the display of the prompt once instead of each time a new data record is merged. The same response is inserted in each resulting merged document.

[*Example*: The following FILLIN field helps the user fill in the correct information by displaying the patient name from the current data record:

```
FILLIN "Please enter the appointment time for MERGEFIELD PatientName :"
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.5.20 FORMCHECKBOX

Syntax:

FORMCHECKBOX

Description: Inserts a check box style form field which, when the editing of form fields is enabled using the documentProtection element (§17.15.1.29), can be checked and unchecked. An instance of this field shall be accompanied by a use of the ffData element (§17.16.17) which contains the form field's properties.

Field Value: A check box based on the properties of the ffData element (§17.16.17).

Switches: None.

[*Example*: Consider the following fields:

FORMCHECKBOX

Assuming the appropriate properties are used in the child XML elements of the field, a check box is displayed.

end example]

17.16.5.21 FORMDROPDOWN

Syntax:

FORMDROPDOWN

Description: Inserts a drop-down list style form field which, when the editing of form fields is enabled using the documentProtection element (§17.15.1.29), can be used to select an entry in the list. An instance of this field shall be accompanied by a use of the ffData element (§17.16.17) which contains the form field's properties.

Field Value: A drop-down list based on the properties of the ffData element (§17.16.17).

Switches: None.

[*Example*: Consider the following fields:

FORMDROPDOWN

Assuming the appropriate properties are used in the child XML elements of the field, a drop-down list is displayed.

end example]

17.16.5.22 FORMTEXT

Syntax:

FORMTEXT

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Description: Inserts a text box style form field which, when the editing of form fields is enabled using the documentProtection element (§17.15.1.29), can be typed into. An instance of this field shall be accompanied by a use of the ffData element (§17.16.17) which contains the form field's properties.

Field Value: A text box based on the properties of the ffData element (§17.16.17).

Switches: None.

[*Example*: Consider the following fields:

```
FORMTEXT
```

Assuming the appropriate properties are used in the child XML elements of the field, a text box is displayed.

end example]

17.16.5.23 GOTOBUTTON

Syntax:

GOTOBUTTON *field-argument-1 field-argument-2*

field-argument-1:
expression

field-argument-2:
expression

Description: Inserts a jump command, such that when it is activated, the insertion point of the document is moved to the location specified by *text* in *field-argument-1*. *text* can be a bookmark, a page number, or some other item (as described below). The page number can be a reference resulting from a REF field. The other items than can be locations are:

a *n*	annotation
f *n*	footnote
l *n*	line
p *n*	page
s *n*	section

where *n* is an integer that designates the *n*th occurrence of the corresponding item (which is not necessarily the item numbered *n*).

text in *field-argument-2* is the text or graphic "button" that appears in the document, such that it can be selected to activate the jump. [*Note*: The BOOKMARK and INCLUDEPICTURE fields make for some interesting button possibilities. *end note*] The text or graphic shall appear on one line in the field result; otherwise, an error occurs.

Field Value: None.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Switches: None.

[*Example*: Consider the following fields:

```
GOTOBUTTON MyBookmark Dest
GOTOBUTTON p3 Page
GOTOBUTTON "f 2" Footnote
```

When the `Dest` "button" is activated, the insertion point becomes the location marked by `MyBookmark`. When `Page` is activated, the insertion point becomes the beginning of the third page. When Footnote is activated, the insertion point becomes the marker of the second footnote, at the place it is used in the document, not in any footnote list.

end example]

17.16.5.24 GREETINGLINE

Syntax:

`GREETINGLINE` [*switches*]

Description: Inserts a mail merge greeting line.

Field Value: The greeting line.

Switches: Zero or more of the following *field-specific-switches*.

`\c` *field-argument*	The *text* in this switch's *field-argument* specifies the text to include in the merge field if the name field in the data source is blank.
`\c` *field-argument*	The *text* in this switch's *field-argument* specifies the format of the name included in the field.
`\l` *field-argument*	The *text* in this switch's *field-argument* specifies the language ID used to format the name. it defaults to the language ID of the first character of the document. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)

17.16.5.25 HYPERLINK

Syntax:

`HYPERLINK` *field-argument* [*switches*]

Description: When selected, causes control to jump to the location specified by *text* in *field-argument*. That location can be a bookmark or a URL.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

`\l` *field-argument*	*text* in this switch's *field-argument* specifies a location in the file,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	such as a bookmark, where this hyperlink jumps.
`\m`	Appends coordinates to a hyperlink for a server-side image map.
`\n`	Causes the destination site to be opened in a new web browser window.
`\o` *field-argument*	*text* in this switch's *field-argument* specifies the ScreenTip text for the hyperlink.
`\t` *field-argument*	*text* in this switch's *field-argument* specifies the target to which the link should be redirected. Use this switch to link from a frames page to a page that you want to appear outside of the frames page. The permitted values for *text* are: • `_top`, whole page (the default) • `_self`, same frame • `_blank`, new web browser window • `_parent`, parent frame

[*Example*:

```
HYPERLINK http://www.example.com/
HYPERLINK "E:\\ReadMe.txt"
```

end example]

17.16.5.26 IF

Syntax:

`IF` *comparison field-argument-1 field-argument-2*

field-argument-1:
expression

field-argument-2:
expression

Description: Compares the values designated by the *expression*s in *comparison* using the operator designated by *comparison-operator*.

[*Note*: *comparison-operator* can be any one of the six relational and equality operators specified for *operator* (§17.16.3.3). *end note*]

If *comparison-operator* is = or <>, the left-hand *expression* operand can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters.

Field Value: *field-argument-1* if the comparison is true; otherwise, *field-argument-2*.

Switches: None.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: The following example specifies that if the customer order is greater than or equal to 100 units, the result is "Thanks"; but if the customer order is fewer than 100 units, the result is "The minimum order is 100 units":

```
{IF order >= 100 "Thanks" "The minimum order is 100 units" }
```

For other examples, see §17.16.2, and the COMPARE field (§17.16.5.10) and the QUOTE field (§17.16.5.48). *end example*]

17.16.5.27 INCLUDEPICTURE

Syntax:

INCLUDEPICTURE *field-argument* [*switches*]

Description: Retrieves the picture named by *field-argument* and displays it as the field result. *Field-argument* specifies the location of the picture using an IRI.

[*Note*: A list of possible image formats is provided in §15.2.14. *end note*]

Field Value: The specified picture.

Switches: Zero or more of the following *field-specific-switches*.

\c *field-argument*	If *text* in this switch's *field-argument* identifies the graphics filter to be used.
\d	Reduce the file size by not storing graphics data with the document.

[*Example*:

```
INCLUDEPICTURE "file:///g:/photos/Ellen%20in%20Oslo.jpg"
```

end example]

17.16.5.28 INCLUDETEXT

Syntax:

INCLUDETEXT *field-argument-1* [*field-argument-2*] [*switches*]

field-argument-1:
field-argument

field-argument-2:
field-argument

Description: Inserts all or part of the text and graphics contained in the document named by *field-argument-1*. *field-argument-1* specifies the location of the document using an IRI (Duerst and Suignard 2005). If the document is a WordprocessingML document, the portion marked by the optional bookmark *field-argument-2* is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

inserted. If no such bookmark is specified here, the whole document is inserted. If the document is an XML file, the fragment referred to by an XPath expression in the \x switch is inserted. If no such switch is specified, the whole XML file is inserted.

If *field-argument-1* contains white space, it shall be enclosed in double quotes.

[*Note*: A list of possible text formats is provided in §11.3.1. *end note*]

Field Value: The specified text and graphics.

Switches: Zero or more of the following *field-specific-switches*.

\!	Prevents this field from being updated unless all fields in the inserted text are first updated in their original document.
\c *field-argument*	Specifies that the file specified by *field-argument-2* shall be processed by a document filter whose name matches the corresponding *field-argument* value. Possible *field-argument* values are implementation-defined.
\e	Specifies the encoding applied to the data within the file referenced by *field-argument-1*. If this argument is omitted, applications should attempt to determine the encoding by reading the file's contents if possible.
\m	Specifies the MIME type of the file referenced by *field-argument-1*, as defined by http://www.iana.org/assignments/media-types/. If this argument is omitted, applications should attempt to determine the file type from its contents if possible.
\n *field-argument*	The *text* in this switch's *field-argument* specifies a namespace mapping for XPath queries. This switch is required if the \x switch refers to an element by name in an XML file that declares a namespace.
\t *field-argument*	The *text* in this switch's *field-argument* specifies an XSLT for formatting XML data.
\x *field-argument*	The *text* in this switch's *field-argument* specifies the XPath for returning a fragment of data in an XML file.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following field inserts the portion of the WordprocessingML document referred to by the bookmark Summary:

```
INCLUDETEXT "file:///C:/Winword/Port Development RFP" Summary
```

The following field inserts the Name element of the XML document Resume.xml and applies the XSLT Display.xsl to it:

```
INCLUDETEXT "file:///C:/Resume.xml" \n xmlns:a=\"resume-schema\"
  \t "file:///C:/display.xsl" \x a:Resume/a:Name
```

 ©ISO/IEC 2016 – All rights reserved

end example]

17.16.5.29 INDEX

Syntax:

INDEX [*switches*]

Description: Builds an index using the index entries specified by XE fields (§17.16.5.72), and inserts that index at this place in the document. Each index entry and subentry is a separate paragraph unless the \r switch is used, in which case, an index entry and all its subentries together make up a paragraph.

Field Value: The index.

Switches: Zero or more of the following *field-specific-switches*.

\b *field-argument*	Builds an index for the portion of the document marked by the bookmark indicated by *text* in this switch's *field-argument*.
\c *field-argument*	Builds an index having the number of columns per page specified by *text* in this switch's *field-argument*. That number can be 1–4. Without this switch, the number of columns is 1.
\d *field-argument*	The *text* in this switch's *field-argument* specifies a sequence of characters that is used to separate sequence numbers and page numbers when the \s switch is used. By default, a hyphen (-) is used.
\e *field-argument*	The *text* in this switch's *field-argument* specifies a sequence of characters that is used to separate an index entry and its first page number. By default, a comma (,) and space sequence is used. If *text* contains a horizontal tab character, the page number list is right justified in the column.
\f *field-argument*	Builds an index using only those entries having the entry type (§17.16.5.72) specified by *text* in this switch's *field-argument*. Without this switch, all entries included.
\g *field-argument*	The *text* in this switch's *field-argument* specifies a sequence of characters that is used to separate the start and end of a page range. By default, an en dash is used.
\h *field-argument*	Builds an index such that the *text* in this switch's *field-argument* occurs as a heading—formatted with the Index Heading style—at the start of each set of entries for any given letter. If the first letter in *text* is A or a, that letter is replaced with the corresponding letter for each letter set. To replace the default heading with a blank line, use a space as *text*. [*Example*: `INDEX \h "a Entries Follow" \c "1" \z "1033"` **F Entries Follow** fox, 1 **Q Entries Follow** quick brown fox. *See* Fox *end example*]
\k *field-argument*	The *text* in this switch's *field-argument* specifies a sequence of characters that is used to separate an index entry and its cross reference (as produced by an XE entry (§17.16.5.72)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	having a \t switch). By default, a period (.) and space sequence is used. [*Example*: The quick brown fox{ XE "fox" } jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox{ XE "quick brown fox" \t "See Fox" } jumps over the lazy dog. The quick brown fox jumps over the lazy dog. INDEX \k <test> \c "1" \z "1033" fox, 1 quick brown fox <test>See Fox *end example*]
\l *field-argument*	The *text* in this switch's *field-argument* specifies a sequence of characters that is used to separate two page numbers in a page number list. By default, a comma (,) and space sequence is used.
\p *field-argument*	Builds an index using only those entries whose first letter is in the range of letters specified by *text* in this switch's *field-argument*. The letter range has the form *startLetter-endLetter*. If *startLetter* is !, entries whose first character is not a letter, are also included, as are the letters starting from A. The letters in the range can be either upper- or lowercase.
\r	Runs subentries into the same line as the main entry. Colons (:) separate main entries from subentries; semicolons (;) separate subentries.
\s *field-argument*	The *text* in this switch's *field-argument* is used as a sequence name, and the sequence number is included along with the page number, these numbers being separated by a hyphen (-), by default. Use the \d switch to specify a separator character other than the default.
\y	Enables the use of yomi text for index entries.
\z *field-argument*	The *text* in this switch's *field-argument* specifies the language ID used to generate the index. This language ID is specified in the format defined by ST_Lang (§22.9.2.6)

[*Example*: The index produced using the corresponding set of index entries and the field `INDEX \c "1" \e "tab" \g " to " \h "A" \z "1033"` is:

B

behavior

implementation-defined **2**

documenting **3**

I

item

package-relationship ..... See package-relationship item

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

O

Office Open XML **2**, *3*, 4

X

XML ..... 1 to 4

end example]

17.16.5.30 KEYWORDS

Syntax:

KEYWORDS [*field-argument*] [*switches*]

Description: Retrieves, and optionally sets, the document's keywords, as recorded in the Keywords element of the Core File Properties part or, if *field-argument* is present, the subject specified by *text* in *field-argument*. Specifying a *field-argument* shall change Keywords to *text*. The Keywords element contains a string of text whose format and semantics is unspecified by ISO/IEC 29500.

Field Value: The document's keywords

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Consider the case in which the Keywords element is as follows:

```
<Keywords>switch, field, syntax</Keywords>
```

and the following field is updated:

KEYWORDS

The result is:

```
switch, field, syntax
```

Updating the following field:

```
KEYWORDS "field, formatting, switch, syntax"
```

causes the Subject element to take on the given value. *end example*]

17.16.5.31 LASTSAVEDBY

Syntax:

LASTSAVEDBY [*switches*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Retrieves the name of the user who last modified and saved the current document, as recorded in the LastModifiedBy element of the Core File Properties part.

Field Value: The name of the user who last modified and saved the current document.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Consider the case in which the LastModifiedBy element is as follows:

```
<LastModifiedBy>Elizabeth Martin</LastModifiedBy>
```

and the following field is updated:

```
LASTSAVEDBY \* Upper
```

the result might be:

```
ELIZABETH MARTIN
```

end example]

17.16.5.32 LINK

Syntax:

`LINK` *field-argument-1 field-argument-2* [*field-argument-3*] [*switches*]

field-argument-1:
field-argument

field-argument-2:
field-argument

field-argument-3:
field-argument

Description: For information copied from another application, this field links that information to its original source file. The application type of the link information is specified by *field-argument-1*. The name and location of the source file is specified by *field-argument-2*. *field-argument-3* specifies the portion of the source file that's being linked. [*Example*: If the source file is a SpreadsheetML document, the reference might be to a cell reference or a named range. For a WordprocessingML document, it might be a bookmark. *end example*]

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\a	Causes this field to be updated automatically.
\b	Inserts the linked object as a bitmap.
\d	Don't store the graphic data with the document, thus reducing the file size.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

\f *field-argument*	Causes the linked object to update its formatting in a particular way, according to the integral value of *text* in this switch's *field-argument*. The possible values are: • 0 Maintain the formatting of the source file • 1 Not supported • 2 Match the formatting of the destination document • 3 Not supported • 4 Maintain the formatting of the source file, if the source file is a SpreadsheetML workbook • 5 Match the formatting of the destination document, if the source file is a SpreadsheetML workbook
\h	Inserts the linked object as HTML format text.
\p	Inserts the linked object as a picture.
\r	Inserts the linked object in rich-text format (RTF).
\t	Inserts the linked object in text-only format.
\u	Inserts the linked object as Unicode text.

[*Example*: The following example inserts a range of cells from a SpreadsheetML worksheet. The \a switch ensures that the information is updated in the WordprocessingML document whenever the worksheet is changed:

```
{ LINK Excel.Sheet.8 "C:\\My Documents\\Profits.xls"
   "Sheet1!R1C1:R4C4" \a \p }
```

end example]

17.16.5.33 LISTNUM

Syntax:

LISTNUM [*field-argument*] [*switches*]

Description: Computes the next value from a specific level of a numbering definition, or a specific valuefrom a specific level of a numbering definition.

The terms numbering definition and level are used as defined in §17.9. Within the context of this field, determining the numbering definition is equivalent to reading the value of the numId element (§17.9.18), and determining the level is equivalent to reading to value of the ilvl element (§17.9.3). [*Rationale*: This field is distinct from the numbering defined by the syntax in §17.9. This syntax stores the numbering as run content, rather than a paragraph property. This allows multiple numbered items to occur in a single paragraph; at locations other than the start of the paragraph, etc. *end rationale*]

For LISTNUM fields, the associated numbering definition is calculated using the first of the following conditions met:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- If *field-argument* is `NumberValue`, `OutlineDefault`, or `LegalDefault`, all items with each value are part of a single numbering definition.
- If the current paragraph has numbering applied, all items within that paragraph are part of its numbering definition.
- If field-argument is present, the field belongs to the numbering definition used by level one of the abstract numbering definition (§17.9.1) whose name element (§17.9.13) matches *field-argument*. If no such list exists, then the field shall behave as though field-argument was omitted.
- If *field-argument* has been omitted, the field belongs to the same numbering definition as the last instance of a `LISTNUM` field or numbered paragraph present in the document.

Once the numbering definition has been determined, the level is calculated as follows:

- If the `\l` switch is present, the value of that switch overrides all subsequent rules.
- If *field-argument* is `NumberValue`, `OutlineDefault`, or `LegalDefault`:
  - The first instance of the field is level one.
  - Each instance of the field which occurs at the start of the paragraph preceded only by other `LISTNUM` fields (or nothing) increments the level by one. [*Example*: A `LISTNUM` field with no field-argument inserted as the first item in a paragraph using level one of a numbering definition is at level two. *end example*]
  - The first instance of the field which occurs after other paragraph content increments the level by one, and all subsequent `LISTNUM` fields belong to the same level. [*Example*: Consider the following paragraph content: `{LISTNUM} {LISTNUM} test {LISTNUM} test2 {LISTNUM} {LISTNUM}`. Within this paragraph, the first two instances of the field are levels one and two, and the third through fifth instances of the field are part of level three. *end example*]
- If *field-argument* has any other value:
  - If the current paragraph has numbering applied, that paragraph numbering resets the level to one.
    - Each `LISTNUM` field which occurs at the start of the paragraph preceded only by other `LISTNUM` fields (or nothing) increments the level by one. [*Example*: A `LISTNUM` field with no field-argument inserted as the first item in a paragraph using level one of a numbering definition is at level two. *end example*]
    - The first `LISTNUM` field which occurs after other paragraph content increments the level by one, and all subsequent `LISTNUM` fields belong to the same level. [*Example*: Consider the following paragraph content: `{LISTNUM} {LISTNUM} test {LISTNUM} test2 {LISTNUM} {LISTNUM}`. Within this paragraph, the first two instances of the field are levels one and two, and the third through fifth instances of the field are part of level three. *end example*]
  - Fields in paragraphs with no numbering applied start at the same level as the last instance of a `LISTNUM` field or numbered paragraph present in the document.

The XML generated for a complex field implementation shall not have the optional field value stored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Although the value of a LISTNUM field is calculated as a consecutive-valued non-negative integer, once a value is computed, the corresponding integer is rendered in the appropriate format, as described below. In all possible display methods, the list entry value is considered to be a number, and the list to be a numbered list. There are nine levels of list, and the result style used for each is determined based on the *field-argument* as follows:

- If the *field-argument* is `NumberValue`, `OutlineDefault`, or `LegalDefault`, the formatting is based on the table below.
- For any other value, the formatting is based on the abstract numbering definition (§17.9.1) whose name element (§17.9.13) matches *field-argument*. If no such list exists, then the field shall behave as though *field-argument* was omitted.
- If *field-argument* has been omitted:
    - If the current paragraph has numbering applied, the field shall take on the format of the next level of the corresponding numbering definition. [*Example*: A `LISTNUM` field inserted into a paragraph using level one of a numbering definition is displayed in the format defined by level two of that numbering definition. *end example*]
    - If not, the field takes on the numbering format of the last instance of a `LISTNUM` field or numbered paragraph present in the document.

The following tables describe the predefined sequences `NumberValue`, `OutlineDefault`, and `LegalDefault`. The formats described are used as defined by ST_NumberFormat (§17.18.59).

Level	NumberValue Value Content and Format
1	The `decimal` format followed immediately by “)”. [*Example*: “3)”, “9)”, and “15)”. *end example*].
2	The `lowerLetter` format followed immediately by “)”. [*Example*: “c)”, “i)”, and “o)”. *end example*].
3	The `lowerRoman` format followed immediately by “)”. [*Example*: “iii)”, “ix)”, and “xv)”. *end example*].
4	The `decimal` format surrounded by “(“ and “)”, without any intervening white space. [*Example*: “(3)”, “(9)”, and “(15)”. *end example*].
5	The `lowerLetter` format surrounded by “(“ and “)”, without any intervening white space. [*Example*: “(c)”, “(i)”, and “(o)”. *end example*].
6	The `lowerRoman` format surrounded by “(“ and “)”, without any intervening white space. [*Example*: “(iii)”, “(ix)”, and “(xv)”. *end example*].
7	The `decimal` format followed immediately by “.”. [*Example*: “3.”, “9.”, and “15.”. *end example*].
8	The `lowerLetter` format followed immediately by “.”. [*Example*: “c.”, “i.”, and “o.”. *end example*].
9	The `lowerRoman` format followed immediately by “.”. [*Example*: “iii.”, “ix.”, and “xv.”. *end example*].

Level	OutlineDefault Value Content and Format
1	The `upperRoman` format followed immediately by “.”. [*Example*: “III.”, “IX.”, and “XV.”. *end example*].

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Level	OutlineDefault Value Content and Format
2	The upperLetter format followed immediately by ".". [*Example*: "C.", "I.", and "O.". *end example*].
3	The decimal format followed immediately by ".". [*Example*: "3.", "9.", and "15.". *end example*].
4	The lowerLetter format followed immediately by ")". [*Example*: "c)", "i)", and "o)". *end example*].
5	The decimal format surrounded by "(" and ")", without any intervening white space. [*Example*: "(3)", "(9)", and "(15)". *end example*].
6	The lowerLetter format surrounded by "(" and ")", without any intervening white space. [*Example*: "(c)", "(i)", and "(o)". *end example*].
7	The lowerLetter format defined by ST_NumberFormat surrounded by "(" and ")", without any intervening white space. [*Example*: "(iii)", "(ix)", and "(xv)". *end example*].
8	The lowerLetter format surrounded by "(" and ")", without any intervening white space. [*Example*: "(c)", "(i)", and "(o)". *end example*].
9	The lowerRoman format surrounded by "(" and ")", without any intervening white space. [*Example*: "(iii)", "(ix)", and "(xv)". *end example*].

Level	LegalDefault Value Content and Format
1	The decimal format followed immediately by ".". [*Example*: "3.", "9.", and "15.". *end example*].
2–9	A sequence consisting of the last number from each previous level (delimited by "." followed immediately by the next value at this level and "." (all in the decimal format). [*Example*: "1.3.", "2.4.9.", and "1.3.7.2.15.". *end example*].

Field Value: The next value, or a specific value, from the numbering definition which this instance of the LISTNUM field belongs to (as described above).

Switches: Zero or more of the following *field-specific-switches*.

\l *field-argument*	*text* in this switch's *field-argument* is an integer that specifies the level in the list, overriding the default behavior of the field. If \l is omitted, for a new series, the default value is 3; otherwise, the current level is continued.
\s *field-argument*	*text* in this switch's *field-argument* is an integer that specifies the starting value for this field. If \s is omitted, for a new series, the default value is 1; otherwise, the current series numbering is continued.

[*Example*: When the following fields are updated:

```
We need to perform the following functions: LISTNUM  NumberDefault \l 3 \s 1 Get approval for the project. LISTNUM Arrange for funding. LISTNUM Hire staff.
```

The results are:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
We need to perform the following functions: i) Get approval for the
project. ii) Arrange for funding. iii) Hire staff.
```

When the following fields are updated:

```
LISTNUM NumberDefault \l 3 \s 1
LISTNUM
LISTNUM NumberDefault
LISTNUM NumberDefault \s 3
LISTNUM
LISTNUM NumberDefault \l 1
LISTNUM
LISTNUM NumberDefault \l 1 \s 1
LISTNUM LegalDefault \1 1 \s 1
LISTNUM LegalDefault
LISTNUM LegalDefault \l 1
LISTNUM LegalDefault \s 4
LISTNUM LegalDefault
```

The results are:

```
i)
ii)
iii)
iii)
iv)
2)
3)
1)
1.1.1.
1.1.2.
2.
2.1.4.
2.1.5.
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.5.34 MACROBUTTON

Syntax:

MACROBUTTON *field-argument-1* *field-argument-2*

field-argument-1:
field-argument

field-argument-2:
field-argument

Description: Allows the macro or command designated by *text* in *field-argument-1* to be run. *text* in *field-argument-2* designates the text or graphic to appear as the "button" that is selected to run the macro or command. The mechanism by which the command specified by *text* in *field-argument-1* is located and/or executed by an application is implementation-defined.

Field Value: *field-argument-2*.

Switches: None.

17.16.5.35 MERGEFIELD

Syntax:

MERGEFIELD *field-argument* [*switches*]

Description: Retrieves the name of a data field designated by *text* in *field-argument* within the merge characters in a mail merge main document. When the main document is merged with the selected data source, information from the specified data field is inserted in place of the merge field.

The name designated by *text* shall match exactly the field name in the header record.

Field Value: The name of a data field designated by *text* in *field-argument*.

Switches: Zero or more of the following *field-specific-switches*.

\b *field-argument*	The *text* in this switch's *field-argument* specifies the text to be inserted before the MERGEFIELD field if the field is not blank.
\f *field-argument*	The *text* in this switch's *field-argument* specifies the text to be inserted after the MERGEFIELD field if the field is not blank.
\m	Specifies that the MERGEFIELD field is a mapped field.
\v	Enables character conversion for vertical formatting.

[*Example*: Placing the following three MERGEFIELD fields together and using the \f switch ensures that the three fields have spaces between them, but only if the designated field information is present in the data source.

```
MERGEFIELD CoutesyTitle \f " " MERGEFIELD FirstName \f " "
  MERGEFIELD LastName
```

end example]

17.16.5.36 MERGEREC

Syntax:

MERGEREC

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: When an application uses the contents of this document to perform a mail merge, this field is an instruction that the *field result* display the number of the corresponding data record in the merged document. The number reflects the sequential order of the data records that were selected and possibly sorted for merging with the active main document. It does not indicate the actual order of the records as they occur in the physical data source. [*Note*: A personnel database might contain thousands of records. However, to send a form letter to employees who've reached their five-year anniversary with your company, you'd select as your data source only the records of those five-year employees, a much smaller set of records. To print a physical record number, you must include a record number field in the data source and insert the corresponding merge field in the main document. *end note*]

Field Value: The literal text «MERGEREC» unless a mail merge is being performed, in which case, the number of the data record being merged.

Switches: None.

[*Example*: The following example uses a MERGEREC field inside a formula to create unique invoice numbers. When the main document is merged with the data source, the number resulting from the MERGEREC field is added to the numbers representing the date and time the invoices are printed.

```
Invoice Number: = { PRINTDATE \@ "MMddyyyyHHmm" + MERGEREC }
```

The result might be:

```
Invoice Number for record 12, printed on Feb. 13, 2003 at 9:46:
  02132003094612
```

end example]

17.16.5.37 MERGESEQ

Syntax:

```
MERGESEQ
```

Description: Counts the number of data records that were successfully merged with the main document. Merged records are numbered starting from 1 each time documents are merged. [*Note*: The number might be different from the value inserted by the MERGEREC field. *end note*]

Field Value: The number of data records that were successfully merged with the main document.

Switches: None.

[*Example*: Consider the case in which only records 10–25 are merged. The MERGESEQ number corresponding to the first data record merged is 1, even though the MERGEREC number for that data record is 10. *end example*]

17.16.5.38 NEXT

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

NEXT

Description: Merges the next data record into the current resulting merged document, rather than starting a new merged document. [*Note*: This field is used when setting up a mailing label and envelope main document during a mail merge. *end note*]

Field Value: None.

Switches: None.

[*Example*: The following fields print three sets of names and phone numbers in each resulting merged document:

```
MERGEFIELD Name MERGEFIELD Phone
NEXT MERGEFIELD Name MERGEFIELD Phone
NEXT MERGEFIELD Name MERGEFIELD Phone
```

end example]

17.16.5.39 NEXTIF

Syntax:

NEXTIF *comparison*

Description: Compares the values designated by the *expressions* in *comparison* using the operator designated by *comparison-operator*. If the comparison is true, the next data record is merged into the current merge document. (Merge fields that follow the NEXTIF in the main document are replaced by values from the next data record rather than the current data record.) If the comparison is false, the next data record is merged into a new merge document.

[*Note*: *comparison-operator* can be any one of the six relational and equality operators specified for *operator* (§17.16.3.3). *end note*]

A NEXTIF fields shall not be used in a footnote, an endnote, annotation, a header, a footer, or a data source. A NEXTIF field shall not be nested within any field.

If *comparison-operator* is = or <>, the right-hand *expression* operand can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters. The expression shall be enclosed in quotation marks so that it is compared as a character string.

Field Value: None.

Switches: None.

17.16.5.40 NOTEREF

Syntax:

NOTEREF *field-argument* [*switches*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Inserts the mark of the footnote or endnote that is marked by the bookmark specified by *text* in *field-argument*.

Field Value: The mark of the footnote or endnote.

Switches: Zero or more of the following *field-specific-switches*.

\f	For a footnote, inserts the reference mark with the same character formatting as the Footnote Reference style. For an endnote, inserts the reference mark with the same character formatting as the Endnote Reference style.
\h	Inserts a hyperlink to the bookmarked endnote or footnote.
\p	Inserts the relative position of the footnote or endnote. If the NOTEREF field occurs before the bookmark, the result is "below". If the NOTEREF field occurs after the bookmark, the result is "above".

[*Example*: Consider the case in which a bookmark called F10 marks the footnote of interest. When the field

```
… (see note { NOTEREF F10 }).
```

is updated, the result might be:

```
… (see note 5).
```

end example]

17.16.5.41 NUMCHARS

Syntax:

NUMCHARS [*switches*]

Description: Retrieves the number of characters in the current document, as recorded in the Characters element of the Application-Defined File Properties part.

Field Value: The number of characters in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

[*Example*: Consider the case in the document has 6,183 words and the following fields are updated:

```
NUMCHARS
NUMCHARS \# #,##0
```

the results are:

```
6183
6,183
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

17.16.5.42 NUMPAGES

Syntax:

NUMPAGES [*switches*]

Description: Retrieves the number of pages in the current document, as recorded in the Pages element of the Application-Defined File Properties part.

Field Value: The number of pages in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

[*Example*: Consider the case in the document has 19 pages and the following fields are updated:

```
NUMPAGES \# #,##0
NUMPAGES \* OrdText
```

the results are:

```
 19
nineteenth
```

end example]

17.16.5.43 NUMWORDS

Syntax:

NUMWORDS [*switches*]

Description: Retrieves the number of words in the current document, as recorded in the Words element of the Application-Defined File Properties part.

Field Value: The number of words in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

[*Example*: Consider the case in the document has 1,243 words and the following fields are updated:

```
NUMWORDS
NUMWORDS \# #,##0
```

the results are:

```
1243
1,243
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.5.44 PAGE

Syntax:

PAGE [*switches*]

Description: Retrieves the number of the current page.

Field Value: The number of the current page.

Switches: Zero or more *general-formatting-switch*es.

[*Example*: When the current page number is 19 and the following fields are updated:

```
PAGE
PAGE \* ArabicDash
PAGE \* ALPHABETIC
PAGE \* roman
```

the results are:

```
19
- 19 -
S
xix
```

end example]

17.16.5.45 PAGEREF

Syntax:

PAGEREF *field-argument* [*switches*]

Description: Inserts the number of the page containing the bookmark specified by *text* in *field-argument* for a cross-reference.

Field Value: The number of the page containing the bookmark.

Switches: Zero or one of the *general-formatting-switch*es, zero or one of the *numeric-formatting-switch*es, and zero or more of the following *field-specific-switches*.

\h	Creates a hyperlink to the bookmarked paragraph.
\p	Causes the field to display its position relative to the source bookmark. If the PAGEREF field is on the same page as the bookmark, it omits "on page #" and returns "above" or "below" only. If the PAGEREF field is not on the same page as the bookmark, the string "on page #" is used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the case in which a bookmark called `Worldpop1990` marks the table containing figures for 1990. When the field

```
The world population in 1991 was 5 billion; for 1990 figures,
see the table { PAGEREF Worldpop1990 \p }.
```

is updated, the position of the table is inserted in place of the field. The result is one of the following::

```
… see the table above.
… see the table below.
… see the table on page 27.
```

end example]

17.16.5.46 PRINT

Syntax:

PRINT *field-argument* [*switches*]

Description: An instruction to send the printer-specific control code characters specified by *text* in *field-argument* to the selected printer when the document is printed.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\p *field-argument-1 field-argument-2*	Allows PostScript strings to be sent to the printer as native PostScript codes. PostScript commands embedded in the document are carried out in the order in which they are inserted. The y coordinate space used for PostScript commands in as follows: The graphics origin (0,0) is in the lower-left corner of the page, and the positive directions are up and to the right. PostScript drawing instructions take place within a drawing rectangle. The graphics origin is translated to the lower-left corner of the drawing rectangle. *text* in this switch's *field-argument-1* defines the drawing rectangle on which the subsequent PostScript instructions operate. *text* in this switch's *field-argument-2* contains the PostScript instructions.

17.16.5.47 PRINTDATE

Syntax:

PRINTDATE [*switches*]

Description: Retrieves the date and time on which the document was last printed, as recorded in the LastPrinted element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

and-time-formatting-switch used is implementation-defined. For a document that has never been printed, the date and time corresponds to `0000-00-00T00:00:00` local time and each text component is XXX.

Field Value: The date and time on which the document was last printed.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri Lunar or Hebrew Lunar calendar, depending on the language specified by the lang element (§17.3.2.20).
\s	Use the Saka Era calendar.

[*Example*: Consider the case in which the LastPrinted element is as follows:

```
<LastPrinted>2006-01-06T19:58:00Z</LastPrinted>
```

and the following fields are updated in a US-English context that is UTC -5:

```
PRINTDATE
PRINTDATE \@ "dddd, MMMM dd, yyyy  HH:mm:ss"
```

the results are:

```
1/6/2006 2:58:00 PM
Friday, January 06, 2006 14:58:00
```

For a document that has never been printed, the result is:

```
0/0/0000 0:00:00 AM
XXX, XXX 00, 0000  00:00:00
```

end example]

17.16.5.48 PRIVATE

Syntax:

```
PRIVATE
```

Description: Provides a private storage area. This field is used to store data for documents converted from other file formats. The field contains data needed for converting a document back to its original file format.

A PRIVATE field is formatted as hidden text.

Field Value: None.

Switches: None.

17.16.5.49 QUOTE

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

QUOTE *field-argument*

Description: Retrieves the text specified by *text* in *field-argument*. This text can include any other fields except SYMBOL.

Field Value: The specified text.

Switches: One or more of the *date-and-time-formatting-switch*, *general-formatting-switch*, or *date-and-time-formatting-switch*es, depending on the type of *field-argument*.

[*Example*: When the current month is January and the following field is updated:

```
Last month was QUOTE IF DATE \@ "M" = 1 "12" "= DATE \@ "M" - 1"/1/2000 \@
"MMMM".
```

the result is:

```
Last month was December.
```

end example]

17.16.5.50 RD

Syntax:

RD *field-argument* [*switches*]

Description: *field-argument* identifies a file to include when creating a table of contents, a table of authorities, or an index using a TOC (§17.16.5.68), TOA (§17.16.5.67), or INDEX field (§17.16.5.29). RD fields that reference a series of files shall be in the same order as the files in the final document. If the location includes a long file name containing spaces, field-argument shall contain delimiting quotes. A single backslash in the file path shall be preceded directly by a backslash.

For a complex field implementation in XML the optional field-value storage is not needed.

Field Value: None.

Switches: One of the following *field-specific-switches*:

\p	Indicates that the path is relative to the current document.

[*Example*: The following fields inserted into one document create a table of contents that includes entries from the three referenced documents:

```
{ TOC }
{ RD C:\\Manual\\Chapters\\Chapter1.doc }
{ RD C:\\Manual\\Chapters\\Chapter2.doc }
{ RD C:\\Manual\\Chapters\\Chapter3.doc }
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.5.51 REF

Syntax:

[REF] *field-argument* [*switches*]

Description: Inserts the text or graphics represented by the bookmark specified by *text* in *field-argument*. The bookmark shall be defined in the current document. Provided the bookmark name is not exactly the same as a field name, the REF prefix can be omitted. If the text marked by the bookmark contains a paragraph mark, the text preceding the REF field assumes the formatting of the paragraph in the bookmark.

Field Value: The specified text or graphics.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper, and zero or one of the following *field-specific-switches*.

\d *field-argument*	*text* in this switch's *field-argument* specifies the character sequence that is used to separate sequence numbers and page numbers.
\f	Increments footnote, endnote, and annotation numbers that are marked by the bookmark, and inserts the corresponding footnote, endnote, and comment text.
\h	Creates a hyperlink to the bookmarked paragraph.
\n	For a referenced paragraph, causes the field result to have the entire paragraph number without trailing periods. No information about prior levels is displayed unless it is included as part of the current level.
\p	Causes the field result to contain the position relative to the source bookmark using the word "above" or "below." If the REF field appears in the document before the bookmark, it evaluates to "below". If the REF field appears after the bookmark, it evaluates to "above". If the REF field appears within the bookmark, an error is returned. This switch can also be used in conjunction with the \n, \r, and \w switches, in which case, "above" or "below" is appended to the end of the field result.
\r	Inserts the entire paragraph number of the bookmarked paragraph in relative context—or relative to its position in the numbering scheme —without trailing periods.
\t	Causes the REF field to suppress non-delimiter or non-numerical text when used in conjunction with the \n, \r, or \w switch.
\w	Inserts the paragraph number of the bookmarked paragraph in full context from anywhere in the document.

[*Example*: The following field

```
REF _Ref116788778 \r \h
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

makes a hyperlink reference. This kind of field is commonly used, to indicate forward references within a document. *end example*]

17.16.5.52 REVNUM

Syntax:

REVNUM

Description: Retrieves the document's revision number (which indicates the number of times the document has been saved), as recorded in the Revision element of the Core File Properties part.

Field Value: The document's revision number.

Switches: None.

[*Example*: Consider the case in which the Revision element is as follows:

```
<Revision>11</Revision>
```

and the following field is updated:

```
REVNUM
```

The result is:

```
11
```

end example]

17.16.5.53 SAVEDATE

Syntax:

SAVEDATE [*switches*]

Description: Retrieves the date and time on which the document was last saved, as recorded in the DateModified element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-and-time-formatting-switch* used is implementation-defined. For a document that has never been saved, the date and time corresponds to `0000-00-00T00:00:00` local time and each text component is XXX.

Field Value: The date and time on which the document was last saved.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri Lunar or Hebrew Lunar calendar, depending on the language specified by the lang element (§17.3.2.20).
\s	Use the Saka Era calendar.

[*Example*: Consider the case in which the DateModified element is as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<DateModified>2006-01-06T20:15:00Z</DateModified>
```

and the following fields are updated in a US-English context that is UTC -5:

```
SAVEDATE
SAVEDATE \@ "dddd, MMMM dd, yyyy  HH:mm:ss"
```

the results are:

```
1/6/2006 3:15:00 PM
Friday, January 06, 2006 15:15:00
```

For a document that has never been saved, the result is:

```
0/0/0000 0:00:00 AM
XXX, XXX 00, 0000  00:00:00
```

end example]

17.16.5.54 SECTION

Syntax:

SECTION [*switches*]

Description: Retrieves the number of the current section.

Field Value: The number of the current section.

Switches: Zero or more *general-formatting-switches*.

[*Example*: When the current section number is 19 and the following fields are updated:

```
SECTION
SECTION \* ArabicDash
SECTION \* ALPHABETIC
SECTION \* roman
```

the results are:

```
19
- 19 -
S
xix
```

end example]

17.16.5.55 SECTIONPAGES

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

SECTIONPAGES [*switches*]

Description: Retrieves the number of the current page within the current section.

Field Value: The number of the current page within the current section.

Switches: Zero or more *general-formatting-switch*es.

[*Example*: When the current page number within the current section is 19 and the following fields are updated:

```
SECTIONPAGES
SECTIONPAGES \* ArabicDash
SECTIONPAGES \* ALPHABETIC
SECTIONPAGES \* roman
```

the results are:

```
19
- 19 -
S
xix
```

end example]

17.16.5.56 SEQ

Syntax:

SEQ *identifier* [*field-argument*] [*switches*]

Description: Sequentially numbers chapters, tables, figures, and other user-defined lists of items in a document. If an item and its SEQ field are added, deleted, or moved, updating the remaining SEQ fields in the document reflects the new sequence. A SEQ field in a header, footer, annotation, or footnote shall not affect the sequence numbering that results from SEQ fields in the document text.

[*Note*: The LISTNUM field also produces automatic numbering and can be a better alternative when creating a complex numbered list. *end note*]

identifier is the name assigned to the series of items that are to be numbered. [*Example*: *identifier* might be Equation, Figure, Table, or Thing, as the user deems appropriate for a caption. *end example*] *identifier* shall start with a Latin letter and shall consist of no more than 40 Latin letters, Arabic digits, and underscores. (See the TOC field (§17.16.5.68) switches \c and \s for uses of *identifier*.)

text in *field-argument* specifies a bookmark name that refers to an item elsewhere in the document rather than in the current location.

Field Value: The next number in the sequence.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Switches: Zero or one of the *numeric-formatting-switch*es, or zero or more of the following *field-specific-switches*. If no *numeric-formatting-switch* is present, `\* Arabic` is used.

`\c`	Repeats the closest preceding sequence number. [*Note*: This is useful for inserting chapter numbers in headers or footers. *end note*]
`\h`	Hides the field result unless a *general-formatting-switch* is also present.[*Note*: This switch can be used to refer to a SEQ field in a cross-reference without printing the number. *end note*]
`\n`	Inserts the next sequence number for the specified item. This is the default.
`\r` *field-argument*	Resets the sequence number to the integer number specified by *text* in this switch's *field-argument*.
`\s` *field-argument*	Resets the sequence number to the built-in (integer) heading level specified by *text* in this switch's *field-argument*.

[*Example*: When the following fields are updated:

```
SEQ Figure
SEQ Figure \* roman
SEQ Figure \n
SEQ Figure \c : …
SEQ Figure \h : …
SEQ Figure
SEQ Figure \r 1
SEQ Figure
```

the results are:

```
Figure 1
Figure ii
Figure 3
Figure 3: …
Figure : …
Figure 5
Figure 10
Figure 11
```

end example]

17.16.5.57 SET

Syntax:

SET *field-argument-1 field-argument-2*

field-argument-1:
field-argument

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

field-argument-2:
field-argument

Description: Defines the bookmark name specified by *field-argument-1* to represent the information specified by *field-argument-2*.

Field Value: None.

Switches: None.

[*Example*: Consider the following:

```
SET EnteredBy "Paul Smith"
SET UnitCost 25.00
SET Quantity FILLIN "Enter number of items ordered:"
SET SalesTax 10%
SET TotalCost = (UnitCost * Quantity) + ((UnitCost * Quantity) * SalesTax)
Total cost: TotalCost \# "$#0.00"
Thank you for your order,
EnteredBy
```

end example]

17.16.5.58 SKIPIF

Syntax:

SKIPIF *comparison*

Description: Compares the values designated by the *expressions* in *comparison* using the operator designated by *comparison-operator*. If the comparison is true, SKIPIF cancels the current merge document, moves to the next data record in the data source, and starts a new merge document. If the comparison is false, the current merge document is continued.

[*Note*: *comparison-operator* can be any one of the six relational and equality operators specified for *operator* (§17.16.3.3). *end note*]

If *comparison-operator* is = or <>, the right-hand *expression* operand can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters. The expression shall be enclosed in quotation marks so that it is compared as a character string.

Field Value: None.

Switches: None.

[*Example*: Inserted into a mail merge main document, the following field examines the contents of the Order field for the current data record. If the field contains a number less than 100, no merged document is produced for that data record.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
SKIPIF MERGEFIELD Order < 100
```

end example]

17.16.5.59 STYLEREF

Syntax:

STYLEREF *field-argument* [*switches*]

Description: Inserts the nearest piece of text prior to this field that is formatted by the style whose name is specified by *text* in *field-argument*. The style can be a paragraph style or a character style.

When this field is used in a header or footer, it results in the first or the last text formatted with the specified style on the current page, allowing for dictionary-style headers or footers.

The location at which a STYLEREF field is inserted determines the direction searched for the style, as follows:

- In document text, by default, the search goes backward from the STYLEREF field. If the style isn't found, the search goes forward from the STYLEREF field.
- In footnotes, annotations, and endnotes, the search goes backward from the footnote, annotation, or endnote reference mark. If the style isn't found, the search goes forward from the reference mark.
- In headers and footers in a printed document, the search is applied to the current page, by default, from top to bottom, for the specified style. If the style isn't found, the search goes from the top of the page to the beginning of the document, and then from the bottom of the page to the end of the document. If the \l switch is used, the search goes from the bottom of the page to the beginning and then to the end of the document.
- In headers and footers in an electronic document, the search goes on in the section that contains the STYLEREF field, from the beginning, for the specified style. If the style isn't found, the search goes from the end of the section to the end of the document.

Field Value: The nearest piece of text prior to this field that is formatted by the style whose name is specified by *text* in *field-argument*.

Switches: Zero or more of the following *field-specific-switches*.

\l	Inserts the nearest text following the field.
\n	Inserts the paragraph number of the referenced paragraph exactly as it appears in the document.
\p	Inserts the relative position of the referenced paragraph as being "above" or "below".
\r	Inserts the paragraph number of the referenced paragraph exactly in relative context.
\t	When used with the \n, \r, or \w switch, causes non-delimiter and non-numerical text to be suppressed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

\w	Inserts the paragraph number of the referenced paragraph in full context, from anywhere in the document.

[*Example*: When the following field is inserted in a header, it displays the contents of the first paragraph formatted with the style "Heading 3" on the current page:

On this page: { STYLEREF "Heading 3" }

To print the first and last names that appear on each page in a membership directory, for example, apply a character style called `Last Name` to each member's last name. Then insert the following fields in the header:

{ STYLEREF "Last Name" } — { STYLEREF "Last Name" \l }

end example]

17.16.5.60 SUBJECT

Syntax:

SUBJECT [*field-argument*] [*switches*]

Description: Retrieves, and optionally sets, the document's subject, as recorded in the Subject element of the Core File Properties part or, if *field-argument* is present, the subject specified by *text* in *field-argument*. Specifying a *field-argument* shall change Subject to *text*.

Field Value: The document's subject.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Consider the case in which the Title element is as follows:

```
<Subject>A specification for fields</Subject>
```

and the following field is updated:

SUBJECT

The result is:

```
A specification for fields
```

Updating the following field:

```
SUBJECT "A specification for WordprocessingML Fields"
```

causes the Subject element to take on the given value. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.5.61 SYMBOL

Syntax:

SYMBOL *field-argument* [*switches*]

Description: Retrieves the character whose code point value is specified in decimal or hexadecimal (by using a leading 0x or 0X) by *text* in *field-argument*. The formatting switches over ride any formatting applied directly to the result.

The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: The specified character.

Switches: Zero or more of the following *field-specific-switches*.

\a	Interprets *text* in *field-argument* as the value of an ANSI character.
\f *field-argument*	Interprets *text* in the switch's *field-argument* as the name of the font from which the character whose value is specified by *text* in the field's *field-argument*. By default, the font used is that for the current text run.
\h	Inserts the symbol without affecting the line spacing of the paragraph. If large symbols are inserted with this switch, text above the symbol might be overwritten.
\j	Interprets *text* in *field-argument* as the value of a Windows-31J character.
\s *field-argument*	Interprets *text* in the switch's *field-argument* as the integral font size in points.
\u	Interprets *text* in *field-argument* as the value of a Unicode character.

[*Example*: Consider the case in which the following fields are updated:

```
SYMBOL 65
SYMBOL 66 \a
SYMBOL 67 \u
SYMBOL 0x20ac \u
SYMBOL 68
SYMBOL 68 \f Symbol
SYMBOL 40 \f Wingdings \s 24
```

the results are:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A
B
C
€
D
Δ
☎

end example]

17.16.5.62 TA

Syntax:

TA [*switches*]

Description: Defines the text and page number for a table of authorities entry, which is used by a TOA field (§17.16.5.67).

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

\b	Applies bold formatting to the page number for the entry. If the table of authorities style for the entry already has bold formatting, \b removes it.
\c *field-argument*	*text* in this switch's *field-argument* specifies the integral entry category, which is a number that corresponds to the order of categories. The number determines how citations are grouped in tables of authorities. If \c is omitted, category 1 is the default.
\i	Applies italic formatting to the page number for the entry. If the table of authorities' style for the entry already has italic formatting, \i removes it.
\l *field-argument*	*text* in this switch's *field-argument* defines the long citation for the entry.
\r *field-argument*	Inserts as the entry's page number the range of pages marked by the bookmark specified by *text* in this switch's *field-argument*.
\s *field-argument*	*text* in this switch's *field-argument* defines the short citation for the entry.

[*Example*: Given the following fields occurring on page 2:

```
TA \l "Hotels v. Leisure Time" \c 2
TA \l "Baldwin v. Alberti, 58 Wn. 2d 243 (1961)"
   \s "Baldwin v. Alberti" \c 1 \b
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

the table of authorities produced by `INDEX \e "tab" \c "1" \z "1033"` is:

Cases

Baldwin v. Alberti, 58 Wn. 2d 243 (1961)…….. 2

Statutes

Hotels v. Leisure Time…… 2

end example]

17.16.5.63 TC

Syntax:

TC *field-argument* [*switches*]

Description: Defines the text and page number for a table of contents (including a table of figures) entry, which is used by a TOC field (§17.16.5.68). The text of the entry is *text* in *field-argument*.

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

`\f` *field-argument*	The type of items collected in a particular contents list. Use a unique list type identifier (typically a letter from A-Z) for each type of list. For example, to build a list of illustrations, mark each illustration with a field such as `TC "Illustration 1" \f i`, where `i` indicates only illustration entries. If no list type is specified, the entry is listed in a table of contents.
`\l` *field-argument*	The level of the TC entry. [*Example*: The field `TC "Entering Data" \l 4` marks a level-4 entry, and applies the built-in style TOC 4 to that entry in the table of contents. *end example*] If no level is specified, level 1 is assumed.
`\n`	Omits the page number for the entry.

17.16.5.64 TEMPLATE

Syntax:

TEMPLATE [*switches*]

Description: The file name of the template used by the current document.

Field Value: The file name of the template used by the current document, as specified by the target of the relationship specified by the id attribute of the attachedTemplate element (§17.15.1.6), or the empty string if no such element exists.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`, and zero or one of the following *field-specific-switches*.

`\p`	Include the full file path name.

[*Example*: Consider the case in which the following fields are updated:

```
TEMPLATE \* Upper
TEMPLATE \p
```

the results might be:

```
NORMAL.DOTM
C:\Templates\Normal.dotm
```

end example]

17.16.5.65 TIME

Syntax:

`TIME` [*switches*]

Description: The current date and time. As specified in Date and Time Formatting (§17.16.4.1), if no *date-and-time-formatting-switch* is present, a date or time result is formatted in an implementation-defined manner.

Field Value: The current date and time.

Switches: Zero or one *date-and-time-formatting-switch*.

[*Example*: Consider the case in which the following fields are updated on the given date and time:

```
TIME
TIME \@ "dddd, MMMM dd, yyyy  HH:mm:ss"
```

the results are:

```
1:59 PM
Friday, January 06, 2006  13:59:42
```

end example]

[*Note*: For some *date-and-time-formatting-switches*, the DATE (§17.16.5.13) and `TIME` fields can produce the same result. *end note*]

17.16.5.66 TITLE

Syntax:

`TITLE` [*field-argument*] [*switches*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Retrieves, and optionally sets, the document's title, as recorded in the Title element of the Core File Properties part or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a *field-argument* shall change Title to *text*.

Field Value: The document's title.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Consider the case in which the Title element is as follows:

```
<Title>My Life's Story</Title>
```

and the following field is updated:

TITLE

The result is:

```
My Life's Story
```

Updating the following field:

```
TITLE "My Life, the Fantasy" \* Upper
```

causes the Title element to take on the value `My Life, the Fantasy` while the result is:

MY LIFE THE FANTASY

end example]

17.16.5.67 TOA

Syntax:

TOA [*switches*]

Description: Builds a table of authorities (that is, a list of the references in a legal document, such as references to cases, statutes, and rules, along with the numbers of the pages on which the references appear) using the entries specified by TA fields (§17.16.5.62).

Field Value: The table of authorities.

Switches: Zero or more of the following *field-specific-switches*.

`\b` *field-argument*	Includes entries only from the portion of the document marked by the bookmark specified by *text* in this switch's *field-argument*.
`\c` *field-argument*	*Includes the entries whose integral category is that specified by text* in this switch's *field-argument*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

\d *field-argument*	Used in conjunction with \s to specify the character sequence that separates the sequence numbers and page numbers. If \d is omitted, a hyphen (-) is used.
\e *field-argument*	*text* in this switch's *field-argument* specifies the character sequence that separates a table of authorities entry and its page number. If \e is not specified, a tab stop with leader dots is used.
\f	Removes the formatting of the entry text in the document from the entry in the table of authorities.
\g *field-argument*	*text* in this switch's *field-argument* specifies the character sequence that separates the pages in a page range. If \g is omitted, an en dash (–) is used.
\h	Includes the category heading for the entries in a table of authorities.
\l *field-argument*	*text* in this switch's *field-argument* specifies the character sequence that separates multiple page references. If \l is omitted, a comma (,) and space are used.
\p	Replaces five or more different page references to the same authority with "passim", which is used to indicate that a word or passage occurs frequently in the work cited.
\s *field-argument*	Includes a case or section number before the page number. The entry shall be numbered with a SEQ field (§17.16.5.56), and *text* in this switch's *field-argument* shall match the identifier in the SEQ field.

[*Example*: See TA (§17.16.5.62). *end example*]

17.16.5.68 TOC

Syntax:

TOC [*switches*]

Description: Builds a table of contents (which can also be a table of figures) using the entries specified by TC fields (§17.16.5.63), their heading levels, and specified styles, and inserts that table at this place in the document. Each table entry is a separate paragraph.

Field Value: The table of contents.

Switches: Zero or more of the following *field-specific-switches*.

\a *field-argument*	Includes captioned items, but omits caption labels and numbers. The identifier designated by *text* in this switch's *field-argument* corresponds to the caption label. Use \c to build a table of captions with labels and numbers.
\b *field-argument*	Includes entries only from the portion of the document marked by

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	the bookmark named by *text* in this switch's *field-argument*.
\c *field-argument*	Includes figures, tables, charts, and other items that are numbered by a SEQ field (§17.16.5.56). The sequence identifier designated by *text* in this switch's *field-argument*, which corresponds to the caption label, shall match the identifier in the corresponding SEQ field.
\d *field-argument*	When used with \s, the *text* in this switch's *field-argument* defines the separator between sequence and page numbers. The default separator is a hyphen (-).
\f *field-argument*	Includes only those TC fields whose identifier exactly matches the *text* in this switch's *field-argument* (which is typically a letter).
\h	Makes the table of contents entries hyperlinks.
\l *field-argument*	Includes TC fields that assign entries to one of the levels specified by *text* in this switch's *field-argument* as a range having the form *startLevel*-*endLevel*, where *startLevel* and *endLevel* are integers, and *startLevel* has a value equal-to or less-than *endLevel*. TC fields that assign entries to lower levels are skipped.
\n *field-argument*	Without *field-argument*, omits page numbers from the table of contents. Page numbers are omitted from all levels unless a range of entry levels is specified by *text* in this switch's *field-argument*. A range is specified as for \l.
\o *field-argument*	Uses paragraphs formatted with all or the specified range of built-in heading styles. Headings in a style range are specified by *text* in this switch's *field-argument* using the notation specified as for \l, where each integer corresponds to the style with a style ID of HeadingX (e.g. 1 corresponds to Heading1). If no heading range is specified, all heading levels used in the document are listed.
\p *field-argument*	*text* in this switch's *field-argument* specifies a sequence of characters that separate an entry and its page number. The default is a tab with leader dots.
\s *field-argument*	For entries numbered with a SEQ field (§17.16.5.56), adds a prefix to the page number. The prefix depends on the type of entry. *text* in this switch's *field-argument* shall match the identifier in the SEQ field.
\t *field-argument*	Uses paragraphs formatted with styles other than the built-in heading styles. *text* in this switch's *field-argument* specifies those styles as a set of comma-separated doublets, with each doublet being a comma-separated set of style name and table of content level. \t can be combined with \o.
\u	Uses the applied paragraph outline level.
\w	Preserves tab entries within table entries.
\x	Preserves newline characters within table entries.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

\z	Hides tab leader and page numbers in web page view (§17.18.102).

[*Example*: The index produced using the corresponding set of index entries and the field `TOC \o "3-3" \h \z \t "Heading 1,1,Heading 2,2,Appendix 1,1,Appendix 2,2,Unnumbered Heading,1"` is:

1. Introduction 1

2. Syntax..... 2

3. XML representation..... 4

4. Formulas and expressions..... 6

4.1 Constants..... 6

4.2 Bookmarks..... 6

4.3 Operators..... 6

4.4 Functions..... 7

4.5 Table cell references..... 8

…

Annex A. Index..... 12

end example]

17.16.5.69 USERADDRESS

Syntax:

USERADDRESS [*field-argument*] [*switches*]

Description: Retrieves the current user's postal address or, if *field-argument* is present, the address specified by *text* in *field-argument*. Specifying a field-argument shall not change the address of the current user.

Field Value: If a particular user's context exists, the current user's postal address; otherwise, an empty string.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Given the current user's address, the following fields:

```
USERADDRESS
USERADDRESS "10 Top Secret Lane, Chiswick" \* Upper
```

produce results of:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
114 Rue du Rhône
CH-1204 Geneva
Switzerland

10 TOP SECRET LANE, CHISWICK
```

end example]

17.16.5.70 USERINITIALS

Syntax:

USERINITIALS [*field-argument*] [*switches*]

Description: Retrieves the current user's initials or, if *field-argument* is present, the initials specified by *text* in *field-argument*. Specifying a field-argument shall not change the initials of the current user.

Field Value: If a particular user's context exists, the current user's initials; otherwise, an empty string.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

[*Example*: Given a current user with initials "DW", the following fields:

```
USERINITIALS \* Lower
USERINITIALS "JaJ"
USERINITIALS "jaj" \* Upper
```

produce results of:

```
dw
JaJ
JAJ
```

end example]

17.16.5.71 USERNAME

Syntax:

USERNAME [*field-argument*] [*switches*]

Description: Retrieves the current user's name or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a field-argument shall not change the name of the current user.

Field Value: If a particular user's context exists, the current user's name; otherwise, an empty string.

Switches: One of the following *general-formatting-switches*: `\* Caps`, `\* FirstCap`, `\* Lower`, or `\* Upper`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Given a current user of "David Williams", the following fields:

```
USERNAME \* Lower
USERNAME "John Jones"
USERNAME "Mary Smith" \* Upper
```

produce results of:

```
david williams
John Jones
MARY SMITH
```

end example]

17.16.5.72 XE

Syntax:

XE *field-argument* [*switches*]

Description: Defines the text and page number for an index entry, which is used by an INDEX field (§17.16.5.29). The text of the entry is *text* in *field-argument*. To indicate a subentry, the main entry text and the subentry text shall be separated by a colon (:). Subentries beyond one level are permitted.

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

\b	Applies bold formatting to the entry's page number. However, if the index style for that entry is already bold, this switch removes that formatting for that entry.
\f *field-argument*	The *text* in this switch's *field-argument* defines an index entry type. If an INDEX field has the same \f switch and *field-argument*, this entry is included in the resulting index; otherwise, it is excluded.
\i	Applies italic formatting to the entry's page number. However, if the index style for that entry is already italic, this switch removes that formatting for that entry.
\r *field-argument*	Instead of the entry's page number, uses the range of pages marked by the bookmark specified by *text* in this switch's *field-argument*.
\t *field-argument*	Uses *text* from *field-argument* in place of a page number. [*Note*: Useful for "See …" or "See also …" entries. *end note*]
\y *field-argument*	Specifies that the *text* from *field-argument* defines the yomi (first phonetic character for sorting indexes) for the index entry.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Given the following fields spread over a series of pages, and a multi-page bookmark called OOXMLPageRange:

```
XE "Office Open XML" \b
XE "syntax" \f "Introduction"
XE "behavior:implementation-defined" \b
XE "Office Open XML" \i
XE "behavior:implementation-defined:documenting" \b
XE "grammar" \f "Introduction" \b
XE "Office Open XML"
XE "item: package-relationship" \t "See package-relationship item"
XE "XML" \r OOXMLPageRange
XE "grammar" \f "Introduction"
XE "production" \f "Introduction"
```

the index produced by `INDEX \e "tab" \c "1" \z "1033"` is:

behavior

implementation-defined **2**

documenting **3**

item

package-relationship ..... See package-relationship item

Office Open XML **2**, 3, 4

XML 1–4

and that produced by `INDEX \f "Introduction" \e "tab" \c "1" \z "1033"` is:

grammar **3**, 5

production 5

syntax 2

end example]

17.16.6 calcOnExit (Recalculate Fields When Current Field Is Modified)

This element specifies that the current contents of all fields within the current WordprocessingML document shall be recalculated from their field codes when the contents of the parent form field are modified. [*Note*: It is at the discretion of an application to determine the scope of a single modification, for example, when the user moves the insertion point in a user interface, or after each keystroke, etc. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If this element is omitted, then modification of the contents of the current field shall not result in all fields in the current document being recalculated.

[*Example*: Consider the following WordprocessingML fragment for the contents of two fields in a document:

```
<w:bookmarkStart w:name="Text1" … />
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:calcOnExit/>
      …
    </w:ffData>
  </w:fldChar>
  </w:r>
  <w:r>
    <w:instrText> FORMTEXT </w:instrText>
  </w:r>
  <w:r>
    <w:fldChar w:fldCharType="separate"/>
  </w:r>
  <w:r>
    <w:t>1</w:t>
  </w:r>
  <w:r>
    <w:fldChar w:fldCharType="end"/>
  </w:r>
  <w:bookmarkEnd … />
  <w:fldSimple w:instr="=Text1+10">
    <w:r>
      <w:t>11</w:t>
    </w:r>
  </w:fldSimple>
```

The first field above (the text form field) has a current value of `1`, but also has the calcOnExit element present (therefore inheriting its default attribute value of `true`). This means that if the value of this form field is changed to `10`, that all fields in the document must automatically be updated, resulting in the second field's value being automatically changed to `20`. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16.7 checkBox (Checkbox Form Field Properties)

This element specifies a set of properties which shall be associated with the parent `FORMCHECKBOX` checkbox form field (§17.16.5.20) within the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If the parent form field is not a checkbox (i.e. its field code does not have a value of FORMCHECKBOX), then these properties can be ignored.

[*Example*: Consider the following WordprocessingML fragment for the properties of a checkbox form field:

```
<w:ffData>
  <w:checkBox>
    <w:size w:val="20" />
    <w:checked w:val="true" />
  </w:checkBox>
</w:ffData>
```

The checkBox element specifies that it contains a set of properties for the parent checkbox form field. In this case, these properties specify that the size of the checkbox must be exactly 10 points via the size element (§17.16.29), and that the current state of the checkbox must be checked via the checked element (§17.16.8). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFCheckBox) is located in §A.1. *end note*]

17.16.8 checked (Checkbox Form Field State)

This element specifies the current state for a checkbox form field. This value shall be used to specify the current value for a checkbox as explicitly chosen for that checkbox, as opposed its default value, which is specified using the default element (§17.16.12).

If this element is omitted, then the parent form field checkbox has no state, and its state shall be determined based on the value of the default element in the checkbox form field properties.

[*Example*: Consider the following WordprocessingML fragment for the properties of a checkbox form field:

```
<w:ffData>
  <w:checkBox>
    …
    <w:checked w:val="true" />
  </w:checkBox>
</w:ffData>
```

The checked element specifies that the current state of the checkbox is checked (via an attribute value of true). *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16.9 ddList (Drop-Down List Form Field Properties)

This element specifies a set of properties which shall be associated with the parent FORMDROPDOWN drop-down list form field (§17.16.5.21) within the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If the parent form field is not a drop-down list (i.e. its field code does not have a value of FORMDROPDOWN), then these properties can be ignored.

[*Example*: Consider the following WordprocessingML fragment for the properties of a drop-down list form field:

```
<w:ffData>
  <w:ddList>
    <w:listEntry w:val="One" />
    <w:listEntry w:val="Two" />
    <w:listEntry w:val="Three" />
  </w:ddList>
</w:ffData>
```

The ddList element specifies that it contains a set of properties for the parent drop-down list form field. In this case, these properties specify that the drop-down list must contain three entries of One, Two, and Three via the listEntry elements (§17.16.25). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFDDList) is located in §A.1. *end note*]

17.16.10 default (Default Text Box Form Field String)

This element specifies the default string for the parent text box form field. This string is the content which shall be displayed in the document story within this form field if its current run contents are empty (i.e. there is not actual content within the text box). If the type (§17.16.34) of the current form field is calculation, then this string shall hold the calculation to be performed.

If this element is omitted, then the current text box form field shall not have a default value.

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

```
<w:fldChar w:fldCharType="begin">
  <w:ffData>
    <w:textInput>
      <w:default w:val="No content."/>
    </w:textInput>
  </w:ffData>
</w:fldChar>
```

The default element specifies the default value of the text box form field to be No content. Since the form field does not contain any value, this is the content which must be displayed when the contents of the form field are displayed by an application. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`
	`  …`
	`</w:sdtPr>`
	
	In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.16.11 default (Default Drop-Down List Item Index)

This element specifies the zero-based index of the default entry for the parent drop-down list form field. This index value is the value within the drop-down list which shall be displayed in the document story within this form field if no element is selected (i.e. the result element (§17.16.28) is omitted).

If this element is omitted, then the current drop-down list form field shall have a default value of `0` (its first entry). If the attribute value references an index value which does not exist (i.e. a negative number or a number that exceeds the number of items in the drop-down list), then this value can be ignored and the current drop-down list form field shall have a default value of `0` (its first entry).

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:fldChar w:fldCharType="begin">
  <w:ffData>
    <w:ddList>
      <w:default w:val="1" />
      <w:listEntry w:val="One" />
      <w:listEntry w:val="Two" />
      <w:listEntry w:val="Three" />
    </w:ddList>
  </w:ffData>
</w:fldChar>
```

The default element specifies the index of the default value of the drop-down list form field to be `1`. Since the form field does not contain a result element, this is the index of the content which must be displayed when the contents of the form field are displayed by an application. In this case, the resulting default value text is `Two`. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.12 default (Default Checkbox Form Field State)

This element specifies the default checkbox state for the parent checkbox form field. This value determines the checkbox state when its current run contents are empty (i.e. there is not actual content within the drop-down list).

If this element is omitted, then the current checkbox form field shall have a default value of `0` (unchecked).

[*Example*: Consider the following WordprocessingML fragment for a checkbox list form field:

 ©ISO/IEC 2016 – All rights reserved

```
<w:fldChar w:fldCharType="begin">
  <w:ffData>
    <w:checkBox>
      <w:default w:val="true" />
    </w:checkBox>
  </w:ffData>
</w:fldChar>
```

The default element specifies the default state of the checkbox form field to be `true`. Since the form field does not contain any run content, this is the state which must be displayed when the contents of the form field are displayed by an application. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16.13 delInstrText (Deleted Field Code)

This element specifies that this run contains deleted field codes (§17.16.5) within a complex field in the document. The delInstrText element shall be used for all runs containing field codes which are part of a region of text that is contained in a deleted region using the del element (§17.13.5.14).

If this element is not contained within a del element, then the document is non-conformant. If this element is contained within a run which is not part of a complex field's field codes, then it should be handled as regular deleted text.

[*Example*: Consider a complex field within a WordprocessingML document which was changed from a text box form field to a checkbox form field with revision tracking enabled. This field would therefore be represented as follows:

```
<w:fldChar w:fldCharType="begin" />
<w:ins>
  <w:r>
    <w:instrText>FORMCHECKBOX</w:instrText>
  </w:r>
</w:ins>
<w:del>
  <w:r>
    <w:delInstrText>FORMFIELDTEXT</w:delInstrText>
  </w:r>
</w:del>
<w:fldChar w:fldCharType="separate" />
…
<w:fldChar w:fldCharType="end" />
```

The deleted field code is contained in a delInstrText node, while the inserted (and current) field code is contained in an instrText node. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules.
	
	[*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`  <w:t>  significant whitespace  </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

17.16.14 enabled (Form Field Enabled)

This element specifies whether the parent form field shall behave as though it is enabled or disabled when it is displayed in the document. This setting shall have no effect on the behavior of this form field unless the document's Settings part specifies that the documentProtection element for the current document is in a state allowing the filling in of form fields.

If this element is omitted, then the parent form field shall be in its enabled state when the document settings specify that the document allows the filling-in of form fields.

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin" >
  <w:ffData>
    <w:enabled w:val="false"/>
    <w:textInput>
      …
    </w:textInput>
  </w:ffData>
</w:fldChar
</w:r>
<w:r>
  <w:instrText> FORMTEXT </w:instrText>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r>
  <w:t>1</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The enabled element specifies that the state of the current text box form field is disabled; therefore this text box must not be editable within the current document even when the state of the documentProtection element specifically allows the editing of form fields. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16.15 entryMacro (Script Function to Execute on Form Field Entry)

This element specifies a subroutine in a scripting language which should be executed when the when the run contents of the parent form field are entered. The language and location of this subroutine can be determined using any method desired by an application. [*Note*: It is at the discretion of an application to determine the scope and timing of "entering" a form field, for example, when the user moves the insertion point in a user interface or upon each operation by an application without a user interface, etc. *end note*]

If this element is omitted, then no subroutine shall be associated with entering the run contents of the parent form field. If this element specifies a macro which cannot be located or is not supported by an application, then its value can be ignored, but shall not be lost upon resaving the file.

[*Example*: Consider the following WordprocessingML fragment for the properties of a checkbox form field:

```
<w:ffData>
  <w:entryMacro w:val="TestEntryFunction" />
  <w:checkBox>
    …
  </w:checkBox>
</w:ffData>
```

The entryMacro element specifies that any application which processes this file should attempt to locate and execute a scripting subroutine called `TestEntryFunction` when the contents of the checkbox are entered. If this subroutine cannot be located or executed, then this setting is silently ignored. *end example*]

Attributes	Description
val (Name of Script Function)	Specifies the name of a single scripting subroutine which shall be associated with the parent element. Its use is specifies based on the context of the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: Consider the following WordprocessingML fragment for the properties of a form field:
	
	`<w:ffData>`
	`  <w:exitMacro w:val="HelloWorld" />`
	`</w:ffData>`
	
	The val attribute specifies that a script function called `HelloWorld` must be used in the context of the parent element; in this case, to execute when the field is exited. *end example*]
	
	The possible values for this attribute are defined by the ST_MacroName simple type (§17.18.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MacroName) is located in §A.1. *end note*]

17.16.16 exitMacro (Script Function to Execute on Form Field Exit)

This element specifies a subroutine in a scripting language which should be executed when the when the run contents of the parent form field are exited. The language and location of this subroutine can be determined using any method desired by an application. [*Note*: It is at the discretion of an application to determine the scope and timing of "exiting" a form field, for example, when the user moves the insertion point in a user interface or upon each operation by an application without a user interface, etc. *end note*]

If this element is omitted, then no subroutine shall be associated with exiting the run contents of the parent form field. If this element specifies a macro which cannot be located or is not supported by an application, then its value can be ignored, but shall not be lost upon resaving the file.

[*Example*: Consider the following WordprocessingML fragment for the properties of a form field:

```
<w:ffData>
  <w:exitMacro w:val="TestExitFunction" />
</w:ffData>
```

The exitMacro element specifies that any application which processes this file should attempt to locate and execute a scripting subroutine called `TestExitFunction` when the contents of the form field are exited. If this subroutine cannot be located or executed, then this setting is silently ignored. *end example*]

Attributes	Description
val (Name of Script Function)	Specifies the name of a single scripting subroutine which shall be associated with the parent element. Its use is specifies based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment for the properties of a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	form field:
	
	`<w:ffData>`
	`  <w:exitMacro w:val="HelloWorld" />`
	`</w:ffData>`
	
	The val attribute specifies that a script function called `HelloWorld` must be used in the context of the parent element; in this case, to execute when the field is exited. *end example*]
	
	The possible values for this attribute are defined by the ST_MacroName simple type (§17.18.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MacroName) is located in §A.1. *end note*]

17.16.17 ffData (Form Field Properties)

This element specifies a set of properties which shall be associated with the parent form field within the document. This form field can be of any of the following types (with the associated field codes in parentheses):

- Checkbox (`FORMCHECKBOX`)
- Drop-down List (`FORMDROPDOWN`)
- Text box (`FORMTEXT`)

If this element is present and the field codes for the document do not specify a form field of one of these types, then the document shall be considered non-conformant.

If this element is omitted, then the properties associated with the parent form field shall be determined based on their default values.

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:name w:val="TextTextBox" />
      <w:enabled w:val="false"/>
      <w:textInput>
        <w:maxLength w:val="10" />
      </w:textInput>
    </w:ffData>
  </w:fldChar>
  </w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:instrText> FORMTEXT </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r>
  <w:t>1</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The ffData element specifies the set of properties for this text box form field; in this example, a form field name of TestTextBox via the name element (§17.16.27), a disabled state via the enabled element (§17.16.14), and a maximum character length of 10 Unicode scalar values via the maxLength element (§17.16.26). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFData) is located in §A.1. *end note*]

17.16.18 fldChar (Complex Field Character)

This element specifies the presence of a complex field character at the current location in the parent run. A *complex field character* is a special character which delimits the start and end of a complex field or separates its field codes from its current field result.

A complex field is defined via the use of the two required complex field characters: a *start character*, which specifies the beginning of a complex field within the document content; and an *end character*, which specifies the end of a complex field. This syntax allows multiple fields to be embedded (or "nested") within each other in a document.

As well, because a complex field can specify both its field codes and its current result within the document, these two items are separated by the optional *separator character*, which defines the end of the field codes and the beginning of the field contents. The omission of this character shall be used to specify that the contents of the field are entirely field codes (i.e. the field has no result).

[*Example*: Consider the following complex field definition within a WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:fldChar w:fldCharType="begin" />
</w:r>
<w:r>
  <w:instrText>AUTHOR</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate" />
</w:r>
<w:r>
  <w:t>Rex Jaeschke</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end" />
</w:r>
```

The three fldChar elements specify:

- The beginning of the field, using the type attribute value of `start`
- The separator between the field codes and the current field results, using the type attribute value of `separate`
- The end of the field, using the type attribute value of end

end example]

If a complex field is not closed before the end of a document story, then no field shall be generated and each individual run shall be processed as if the field characters did not exist (i.e. the contents of all field code run content shall not be displayed, and the field results shall be displayed as literal text).

[*Example*: Consider the following WordprocessingML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:body>
  <w:p>
    <w:r>
      <w:fldChar w:fldCharType="begin" />
    </w:r>
    <w:r>
      <w:instrText>AUTHOR</w:instrText>
    </w:r>
    <w:r>
      <w:fldChar w:fldCharType="separate" />
    </w:r>
    <w:r>
      <w:t>Rex Jaeschke</w:t>
    </w:r>
  </w:p>
</w:body>
```

The complex field is technically incorrect since no end character exists in the main document story. The resulting content must be interpreted as though no field characters exist, resulting in only the literal text `Rex Jaeschke` being displayed in the document. *end example*]

Attributes	Description

dirty (Field Result Invalidated)	Specifies that this field has been flagged by an application to indicate that its current results are no longer correct (stale) due to other modifications made to the document, and these contents should be updated before they are displayed if this functionality is supported by the next processing application.
	
	[*Rationale*: This functionality allows applications with limited subsets of the full functionality of ISO/IEC 29500 to process Word Open XML documents without needing to understand and update all fields based on the semantics for their field codes.
	
	For example, an application can add a new paragraph and flag the table of contents as dirty, without needing to understand anything about how to recalculate that field's content. *end rationale*]
	
	If this attribute is omitted, then its value shall be assumed to be `false`. If the type of the current field character is not `start`, then his setting can be ignored.
	
	[*Example*: Consider the following WordprocessingML for a complex field:
	
	`…`
	`<w:r>`
	`  <w:fldChar w:fldCharType="begin" w:dirty="true"/>`
	`</w:r>`
	`<w:r>`
	`  <w:instrText>TOC /l 1-3</w:instrText>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</w:r> <w:r> <w:fldChar w:fldCharType="separate"/> </w:r> … The dirty attribute value of `true` specifies that the contents of this field are no longer current based on the contents of the document, and should be recalculated whenever an application with this functionality reads the document. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
fldCharType (Field Character Type)	Specifies the type of the current complex field character in the document. [*Example*: Consider the following WordprocessingML for a complex field character: … <w:fldChar w:fldCharType="separate" /> … The type attribute value of `separate` specifies that this is a complex field separator character; therefore it is being used to separate the field codes from the field contents in a complex field. *end example*] The possible values for this attribute are defined by the ST_FldCharType simple type (§17.18.29).
fldLock (Field Should Not Be Recalculated)	Specifies that the parent complex field shall not have its field result recalculated, even if an application attempts to recalculate the results of all fields in the document or a recalculation is explicitly requested. If this attribute is omitted, then its value shall be assumed to be `false`. If the type of the current field character is not `start`, then his setting can be ignored. [*Example*: Consider the following WordprocessingML for a complex field: <w:r> <w:fldChar w:fldCharType="begin" w:fldLock="true"/> </w:r> … <w:r> <w:fldChar w:fldCharType="separate"/> </w:r> <w:r> <w:t>field result</w:t> </w:r> <w:r> <w:fldChar w:fldCharType="end" />

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	`</w:r>` The fldLock attribute value of `true` specifies that the contents of this field must remain `field result` regardless of the actual result of the current field codes. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FldChar) is located in §A.1. *end note*]

17.16.19 fldSimple (Simple Field)

This element specifies the presence of a simple field at the current location in the document. The semantics of this field are defined via its field codes (§17.16.5).

[*Example*: Consider the following WordprocessingML fragment for a simple field:

```
<w:fldSimple w:instr="FILENAME">
  <w:r>
    <w:t>Example Document.docx</w:t>
  </w:r>
</w:fldSimple>
```

The fldSimple element defines a `FILENAME` field (§17.16.5.17) using the simple field syntax. The current field result for the field is `Example Document.docx`. *end example*]

Attributes	Description
dirty (Field Result Invalidated)	Specifies that this field has been flagged by an application to indicate that its current results are no longer correct (stale) due to other modifications made to the document, and these contents should be updated before they are displayed if this functionality is supported by the next processing application. [*Rationale*: This functionality allows applications with limited subsets of the full functionality of ISO/IEC 29500 to process Word Open XML documents without needing to understand and update all fields based on the semantics for their field codes. For example, an application can add a new paragraph and flag the table of contents as dirty, without needing to understand anything about how to recalculate that field's content. *end rationale*] If this attribute is omitted, then its value shall be assumed to be `false`. [*Example*: Consider the following WordprocessingML for a simple field: `<w:fldSimple w:instr="AUTHOR" w:dirty="true"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The dirty attribute value of `true` specifies that the contents of this field are no longer current based on the contents of the document, and should be recalculated whenever an application with this functionality reads the document. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

fldLock (Field Should Not Be Recalculated)	Specifies that the parent field shall not have its field result recalculated, even if an application attempts to recalculate the results of all fields in the document or a recalculation is explicitly requested.
	
	If this attribute is omitted, then its value shall be assumed to be `false`.
	
	[*Example*: Consider the following WordprocessingML for a simple field:
	
	`<w:fldSimple w:instr="AUTHOR" w:fldLock="true">`
	`  <w:r>`
	`  <w:t>Rex Jaeschke</w:t>`
	`  </w:r>`
	`</w:fldSimple>`
	
	The fldLock attribute value of `true` specifies that the contents of this field must remain `Rex Jaeschke` regardless of the actual result of the current field codes. *end example*]
	
	The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

instr (Field Codes)	Specifies the field codes for the simple field. The possible field codes are defined in §17.16.5.
	
	[*Example*: Consider the following WordprocessingML for a simple field:
	
	`<w:fldSimple w:instr="AUTHOR" w:fldLock="true">`
	`  <w:r>`
	`  <w:t>Rex Jaeschke</w:t>`
	`  </w:r>`
	`</w:fldSimple>`
	
	The instr attribute specifies the field codes for this simple field to be `AUTHOR`. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SimpleField) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.16.20 format (Text Box Form Field Formatting)

This element specifies the field formatting which shall be applied to the contents of the parent form field whenever those contents are modified. The type of formatting which is applied to the field depends on the value of its type element (§17.16.34), as follows:

- When the type is equal to `currentDate`, `currentTime`, or `date`, a date formatting string using the syntax defined in §17.16.4.1
- When the type is equal to `calculated` or `number`, a number formatting string using the syntax defined in §17.16.4.2
- When the type is equal to regular, a text formatting string defined as follows:

Argument	Description
`Uppercase`	All letters are uppercase. [*Example*: `Mary Smith` results in `MARY SMITH`. *end example*]
`Lowercase`	All letters are lowercase. [*Example*: `Mary Smith` results in `mary smith`. *end example*]
`First capital`	Capitalizes the first letter of the first word. [*Example*: `Mary Smith` results in `Mary smith`. *end example*]
`Title case`	Capitalizes the first letter of each word. [*Example*: `Mary Smith` results in `Mary Smith`. *end example*]

[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:

```
<w:ffData>
  <w:textInput>
    <w:type w:val="number" />
    <w:maxLength w:val="4" />
    <w:format w:val="0.00" />
  </w:textInput>
</w:ffData>
```

The format element specifies the field formatting which is applied to the input to the field (in this case, a grouping of number formatting picture items as the type element specifies a value of `number`). If a value of 8 was entered into this field, the formatted result would be `8.00`. *end example*]

Attributes	Description
val (String Value)	Specifies that its contents contain a string. The contents of this string are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:pPr> <w:pStyle w:val="Heading1" /> </w:pPr> The value of the val attribute is the ID of the associated paragraph style's styleId. However, consider the following fragment: <w:sdtPr> <w:alias w:val="SDT Title Example" /> … </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.16.21 helpText (Associated Help Text)

This element specifies optional help text which shall be associated with the parent form field. The method or user interface by which this help text can be surfaced is not defined by ISO/IEC 29500.

If this element is omitted, then no help text shall be associated with the current form field.

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:helpText w:type="text" w:val="Example help text."/>
      <w:ddList>
      …
      </w:ddList>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMDROPDOWN </w:instrText>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The helpText element specifies the help text for the parent form field - in this case, literal help text consisting of the string `Example help text.` *end example*]

Attributes	Description

type (Help Text Type)	Specifies the type of help text which is specified by this element, defined by the simple type below.
	
	If this attribute is omitted, then its value shall be assumed to be `text`.
	
	[*Example*: Consider the following WordprocessingML fragment for a form field:
	
	`<w:ffData>`
	`  <w:helpText w:type="text" w:val="Example help text." />`
	`</w:ffData>`
	
	The type attribute has a value of `text`, which specifies that the text in the val attribute is the literal help text for this form field. *end example*]
	
	The possible values for this attribute are defined by the ST_InfoTextType simple type (§17.18.43).

val (Help Text Value)	Specifies the help text for the current form field. Based on the value of the type attribute, the contents of this field shall be interpreted as follows:
	• When the type attribute value is `text`, contains the literal help text for the form field.
	• When the type attribute value is `autoText`, contains the name of a glossary document entry which contains the help text for the form field.
	
	[*Example*: Consider the following WordprocessingML fragment for a form field:
	
	`<w:ffData>`
	`  <w:helpText w:type="autoText" w:val="HelpText" />`
	`</w:ffData>`
	
	The text in the val attribute is the name of a glossary document entry containing the help text for this form field, since the type attribute has a value of `autoText`. *end example*]
	
	The possible values for this attribute are defined by the ST_FFHelpTextVal simple type (§17.18.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFHelpText) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.22 hyperlink (Hyperlink)

This element specifies the presence of a hyperlink at the current location in the document.

[*Example*: Consider the following WordprocessingML fragment for a hyperlink:

```
<w:hyperlink r:id="rId10">
  <w:r>
    <w:t>Click here</w:t>
  </w:r>
</w:hyperlink>
```

The hyperlink element defines a hyperlink whose display text is `Click here`, and whose target is specified by the relationship with an Id attribute value of `rId10`. *end example*]

Attributes	Description

anchor (Hyperlink Anchor)	Specifies the name of a bookmark in the current document which shall be the target of this hyperlink.
	
	If this attribute is omitted, then the default behavior shall be to navigate to the start of the document. If a hyperlink target is also specified using the r:id attribute, then this attribute shall be ignored. If no bookmark exists in the current document with the given bookmark name, then the default behavior shall be to navigate to the start of the document.
	
	[*Example*: Consider the following WordprocessingML fragment for a hyperlink:
	
	`<w:hyperlink w:anchor="chapter3">`
	`  <w:r>`
	`  <w:t>Go to Chapter Three</w:t>`
	`  </w:r>`
	`</w:hyperlink>`
	
	The anchor attribute specifies that the target of the current hyperlink must be the text contained within the bookmark `chapter3` within the document. *end example*]
	
	The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

docLocation (Location in Target Document)	Specifies a location in the target of the hyperlink that has no bookmarks. The method by which the contents of this attribute are linked to document text is outside the scope of ISO/IEC 29500. If this attribute is omitted, then no location shall be associated with the parent hyperlink. If the anchor attribute is also specified, then this attribute can be ignored when the hyperlink is invoked. [*Example*: Consider the following WordprocessingML fragment for a hyperlink:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<w:hyperlink r:id="rId9" w:docLocation="table"> <w:r> <w:t>Click Here</w:t> </w:r> </w:hyperlink> The docLocation attribute specifies that the target of the current hyperlink must be a region targeted by the string table within the target document. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
history (Add To Viewed Hyperlinks)	Specifies whether the target of the parent hyperlink (as specified via the r:id attribute) shall be added to a list of viewed hyperlinks when it is invoked. If this attribute is omitted, then its value shall be assumed to be false. [*Example*: Consider the following WordprocessingML fragment for a hyperlink: <w:hyperlink r:id="rId9" w:history="true"> <w:r> <w:t>http://www.example.com</w:t> </w:r> </w:hyperlink> The history attribute value of true specifies that the target of the current hyperlink must be added to a list of visited hyperlinks when invoked within the document. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).
id (Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the ID of the relationship whose target shall be used as the target for this hyperlink. If this attribute is omitted, then there shall be no external hyperlink target for the current hyperlink - a location in the current document can still be target via the anchor attribute. If this attribute exists, it shall supersede the value in the anchor attribute. [*Example*: Consider the following WordprocessingML fragment for a hyperlink: <w:hyperlink r:id="rId9"> <w:r> <w:t>http://www.example.com</w:t> </w:r> </w:hyperlink> The id attribute value of rId9 specifies that relationship in the associated relationship part item with a corresponding Id attribute value must be navigated to when this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>hyperlink is invoked. For example, if the following XML is present in the associated relationship part item:

<pre><Relationships xmlns="…">
  <Relationship Id="rId9" Mode="External"
Target=http://www.example.com />
</Relationships></pre>
The target of this hyperlink would therefore be the target of relationship rId9 - in this case, http://www.example.com. end example]

The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).</td></tr>
<tr><td>tgtFrame (Hyperlink Target Frame)</td><td>Specifies a frame within the parent HTML frameset for the target of the parent hyperlink when one exists. All values specified by this element shall be handled as follows:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>_top</td><td>Open hyperlink target in the full region of the current window.</td></tr>
<tr><td>_self</td><td>Open hyperlink target in the same frame as the hyperlink appears.</td></tr>
<tr><td>_parent</td><td>Open hyperlink target in the parent of the current frame, or the current frame if this frame has no parent.</td></tr>
<tr><td>_blank</td><td>Open hyperlink target in a new web browser window.</td></tr>
<tr><td>all other values</td><td>Open hyperlink target in the frame with the specified name. If no frame exists with this name, open in the current frame.

If this string does not begin with an alphabetic character, it shall be ignored.</td></tr>
</table>

If this attribute is omitted, then no target frame information shall be associated with the parent hyperlink. If the current document is not part of a frameset, then this information can be ignored.

[Example: Consider the following WordprocessingML fragment for a hyperlink:</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<w:hyperlink r:id="rId9" w:tgtFrame="_top"> <w:r> <w:t>http://example.com</w:t> </w:r> </w:hyperlink> The tgtFrame attribute value of _top specifies that the target of this hyperlink must be displayed in the full extents of the current window. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
tooltip (Associated String)	Specifies a string which can be surfaced in a user interface as associated with the parent hyperlink. The method by which this string is surfaced by an application is outside the scope of ISO/IEC 29500. If this attribute is omitted, then no associated string shall be linked to the parent hyperlink in the document. [*Example*: Consider the following WordprocessingML fragment for a hyperlink: <w:hyperlink r:id="rId9" w:tooltip="Click here!"> <w:r> <w:t>http://example.com</w:t> </w:r> </w:hyperlink> The tooltip attribute value specifies that the parent hyperlink has the associated string of Click here!, which can be used as desired. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlink) is located in §A.1. *end note*]

17.16.23 instrText (Field Code)

This element specifies that this run contains field codes (§17.16.5) within a complex field in the document.

If this element is contained within a run which is not part of a complex field's field codes, then it and its contents should be treated as regular text. If this element is contained within a del element, then the document is non-conformant.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a complex checkbox field within a WordprocessingML. This field would be represented as follows:

```
<w:r>
  <w:fldChar w:fldCharType="begin" />
</w:r>
<w:r>
  <w:instrText>FORMCHECKBOX</w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate" />
</w:r>
…
<w:r>
  <w:fldChar w:fldCharType="end" />
</w:r>
```

The field code is contained in an instrText node which occurs within the field codes portion of the complex field (i.e. before the separator character). *end example*]

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules.
	
	[*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`  <w:t>  significant whitespace  </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.1. *end note*]

17.16.24 label (Form Field Label)

This element specifies the label identifier associated with the current form field. The identifier representing the label shall be stored on this element's val attribute and is used to reference the unique identifier value of a structured document tag. The contents of the structured document tag resolved by a specific unique identifier shall be used as the label content for the form field that references that specific unique identifier of the structured document tag. If multiple instances of the label element are present, the labels referenced are

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

ordered from most general to most specific. [*Example*: A form element for specifying country name might reference the label for these three items (in order): “Sender”, “Home Address”, and “Country”. *end example*]

If this element is omitted or the value of the label identifier cannot be resolved, then no label shall be associated with the given form field.

[*Example*: Consider the following text box form field, which references a structured document tag as a label:

```
<w:sdt>
  <w:sdtPr>
    <w:id w:val="5" />
  </w:sdtPr>
  <w:sdtContent>
    <w:p>
      <w:r>
        <w:t>Name</w:t>
      </w:r>
    </w:p>
  </w:sdtContent>
</w:sdt>
…
<w:ffData>
  <w:name w:val="TextTextBox" />
  <w:enabled />
  <w:textInput />
  <w:label w:val="5" />
</w:ffData>
```

The label element specifies that the text box form field uses the contents of the structured document tag with an identifier value of 5 as a label source. In this example, the label contents is “Name”. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.16.25 listEntry (Drop-Down List Entry)

This element specifies the presence of a single drop-down list entry within the parent drop-down list form field in the document. The order of appearance of the series of listEntry elements in the WordprocessingML markup shall dictate the order of the entries in the drop-down list when it is displayed.

[*Example*: Consider the following WordprocessingML fragment for the properties of a drop-down list form field:

```
<w:ffData>
  <w:ddList>
    <w:listEntry w:val="One" />
    <w:listEntry w:val="Two" />
    <w:listEntry w:val="Three" />
  </w:ddList>
</w:ffData>
```

The three listEntry elements each specify one drop-down list entry for the parent drop-down list form field. In this case, these properties specify that the drop-down list must contain three entries of `One`, `Two`, and `Three` in that order when displayed. *end example*]

Attributes	Description

val (String Value)	Specifies that its contents contain a string.
	
	The contents of this string are interpreted based on the context of the parent XML element.
	
	[*Example*: Consider the following WordprocessingML fragment:
	
	`<w:pPr>`
	`  <w:pStyle w:val="Heading1" />`
	`</w:pPr>`
	
	The value of the val attribute is the ID of the associated paragraph style's `styleId`.
	
	However, consider the following fragment:
	
	`<w:sdtPr>`
	`  <w:alias w:val="SDT Title Example" />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </w:sdtPr> In this case, the decimal number in the val attribute is the caption of the nearest ancestor structured document tag. In each case, the value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.1. *end note*]

17.16.26 maxLength (Text Box Form Field Maximum Length)

This element specifies the maximum length of text which should be allowed within the parent text box form field before any formatting specified by the format element (§17.16.20). If the current contents of this field exceed the specified value when the document is loaded, that violation shall not result in an error, but the application shall prevent the addition of any additional characters until the contents are brought below that limit.

If this element is omitted, then there shall be no limit on the number of characters in the parent text box form field.

[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:

```
<w:ffData>
  <w:textInput>
    <w:type w:val="number" />
    <w:maxLength w:val="4" />
    <w:format w:val="0.00" />
  </w:textInput>
</w:ffData>
```

The maxLength element specifies that the contents of this form field should not be allowed to exceed four characters when edited by an application. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… w:val="1512645511" /> The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

17.16.27 name (Form Field Name)

This element specifies the name of the current form field.

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:name w:val=”FirstName” />
      <w:textInput>
        …
      </w:textInput>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMTEXT </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r>
  <w:t>1</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The name element specifies that the name of the current form field is `FirstName`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

val (Form Field Name Value)	Specifies the name of the form field.
	
	If this attribute is omitted, then the parent form field shall have no name.
	
	[*Example*: Consider the following WordprocessingML fragment for a form field:
	
	`<w:ffData>`
	`  <w:name w:val="ExampleFieldName"/>`
	`</w:ffData>`
	
	The val attribute specifies that the name of the current form field is `ExampleFieldName`. *end example*]
	
	The possible values for this attribute are defined by the ST_FFName simple type (§17.18.26).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFName) is located in §A.1. *end note*]

17.16.28 result (Drop-Down List Selection)

This element specifies the zero-based index of the currently selected entry for the parent drop-down list form field.

If this element is omitted, then the current drop-down list form field shall have the selection specified by the value of the default element (§17.16.11). If the attribute value references an index value which does not exist (i.e. a negative number or a number that exceeds the number of items in the drop-down list), then this value can be ignored and the current drop-down list form field shall have the selection specified by the value of the default element.

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:ddList>
        <w:default w:val="1" />
        <w:result w:val="2" />
        <w:listEntry w:val="One" />
        <w:listEntry w:val="Two" />
        <w:listEntry w:val="Three" />
      </w:ddList>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMDROPDOWN </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The result element specifies the index of the currently selected value of the drop-down list form field to be 2. In this case, the resulting default value text is Three. *end example*]

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute contains a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of simple type ST_DecimalNumber: `<… w:val="1512645511" />` The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§17.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DecimalNumber) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.16.29 size (Checkbox Form Field Size)

This element specifies the exact size for the parent checkbox form field. The resulting field shall be displayed in this point size regardless of the size specified by the formatting of its corresponding content in the document via the style hierarchy.

[*Example*: Consider the following WordprocessingML fragment for the properties of a checkbox form field:

```
<w:ffData>
  <w:checkBox>
    <w:size w:val="20" />
    <w:checked w:val="true" />
  </w:checkBox>
</w:ffData>
```

The size element specifies that the checkbox must be displayed in a ten point font size, regardless of the formatting which would normally be applied to this text via the style hierarchy. *end example*]

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. [*Example*: Consider the following WordprocessingML fragment: `<… w:val="30" />` The value in the val attribute is 30, which is equivalent to 15 points (30 half-points). This value is interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_HpsMeasure simple type (§17.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HpsMeasure) is located in §A.1. *end note*]

17.16.30 sizeAuto (Automatically Size Form Field)

This element specifies that the parent checkbox form field shall be formatted using the point size which is applied to its field characters via the style hierarchy.

[*Example*: Consider the following WordprocessingML fragment for the properties of a checkbox form field:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:r>
  <w:rPr>
    <w:sz w:val="40"/>
  </w:rPr>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:checkBox>
        <w:sizeAuto />
        <w:checked w:val="true" />
      </w:checkBox>
    </w:ffData>
  </w:fldChar>
</w:r>
```

The sizeAuto element specifies that the checkbox must be displayed in the point size of the formatting which would normally be applied to this text via the style hierarchy. In this case, this size is the twenty points specified via the direct formatting on the parent run. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.16.31 statusText (Associated Status Text)

This element specifies optional status text which shall be associated with the parent form field. The method or user interface by which this status text can be surfaced is not defined by ISO/IEC 29500.

If this element is omitted, then no status text shall be associated with the current form field.

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:statusText w:type="text" w:val="Example status text."/>
      <w:ddList>
        …
      </w:ddList>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMDROPDOWN </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The statusText element specifies the status text for the parent form field - in this case, literal text consisting of the string `Example status text.` *end example*]

Attributes	Description

type (Status Text Type)	Specifies the type of status text which is specified by this element, defined by the simple type below.
	
	If this attribute is omitted, then its value shall be assumed to be `text`.
	
	[*Example*: Consider the following WordprocessingML fragment for a form field:
	
	`<w:ffData>`
	`  <w:statusText w:type="text" w:val="Example status text." />`
	`</w:ffData>`
	
	The type attribute has a value of `text`, which specifies that the text in the val attribute is the literal status text for this form field. *end example*]
	
	The possible values for this attribute are defined by the ST_InfoTextType simple type (§17.18.43).

val (Status Text Value)	Specifies the status text for the current form field. Based on the value of the type attribute, the contents of this field shall be interpreted as follows:
	• When the type attribute value is `text`, contains the literal status text for the form field.
	• When the type attribute value is `autoText`, contains the name of a glossary document entry which contains the status text for the form field.
	
	[*Example*: Consider the following WordprocessingML fragment for a form field:
	
	`<w:ffData>`
	`  <w:statusText w:type="autoText" w:val="MyStatusText" />`
	`</w:ffData>`
	
	The text in the val attribute is the name of a glossary document entry containing the status text for this form field, since the type attribute has a value of `autoText`. *end example*]
	
	The possible values for this attribute are defined by the ST_FFStatusTextVal simple type (§17.18.27).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFStatusText) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.16.32 tabIndex (Form Field Navigation Order Index)

This element specifies the position of the current form field in the navigation (tab) order used in the document. The tabbing index shall be stored on this element's val attribute and is analogous to the tabIndex attribute in HTML.

Objects that support tab index shall be navigated by consumers in the following order:

- Objects for which the XML specifies a non-zero tabIndex value are navigated first. Navigation proceeds with the element with the lowest resolved value of tabIndex to the element with the highest resolved value of tabIndex.
    - Objects that specify identical resolved values of tabIndex is navigated in the lexical order in which the elements appear in the underlying WordprocessingML.
- Objects for which the XML does not specify an index or objects for which the XML specifies a resolved tabIndex value of 0 are navigated last. These objects are navigated in the lexical order in which they appear in the underlying WordprocessingML.

[*Example*: Consider the following two text box form fields where form field specifies a tab index:

```
<w:ffData>
  <w:name w:val="FirstName" />
  <w:enabled />
  <w:textInput />
  <w:tabIndex w:val="1" />
</w:ffData>
…
<w:ffData>
  <w:name w:val="LastName" />
  <w:enabled />
  <w:textInput />
  <w:tabIndex w:val="2" />
</w:ffData>
```

The tabIndex element specifies that the `FirstName` form field must be the first content to be reached via tabbing, whereas the `LastName` form field must be the second content to be reached via tabbing. *end example*]

Attributes	Description
val (Positive Decimal Number Value)	Specifies that the contents of this attribute contains a positive decimal number. The contents of this positive decimal number are interpreted based on the context of the parent XML element. [*Example*: Consider the following numeric WordprocessingML property of type ST_UnsignedDecimalNumber:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… w:val="15" /> The value of the val attribute is a decimal number whose value must be interpreted in the context of the parent element. *end example*] The possible values for this attribute are defined by the ST_UnsignedDecimalNumber simple type (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedDecimalNumber) is located in §A.1. *end note*]

17.16.33 textInput (Text Box Form Field Properties)

This element specifies a set of properties which shall be associated with the parent FORMTEXT text box form field (§17.16.5.22) within the document.

If the parent form field is not a text box (i.e. its field code does not have a value of FORMTEXT), then these properties can be ignored.

[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:

```
<w:ffData>
  <w:textInput>
    <w:maxLength w:val="4" />
    …
    <w:type w:val="number" />
  </w:textInput>
</w:ffData>
```

The textInput element specifies that it contains a set of properties for the parent text box form field. In this case, these properties specify that the drop-down list must contain no more than four characters via the maxLength element (§17.16.26), and that its contents must contain a number via the type element (§17.16.34). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFTextInput) is located in §A.1. *end note*]

17.16.34 type (Text Box Form Field Type)

This element specifies the type of the contents of the current text box form field. This element shall not be used to prevent the successful loading of any contents in the field, but shall be used to parse the formatting specified in the format element (§17.16.20) and should be used to prevent the addition of illegal content when its contents are edited by an application.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If this element is omitted, then its default value shall be assumed to be `regular`.

[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:

```
<w:ffData>
  <w:textInput>
    <w:type w:val="number" />
    <w:maxLength w:val="4" />
    <w:format w:val="0.00" />
  </w:textInput>
</w:ffData>
```

The type element specifies that the contents of this form field should be handled as a `number` by an application. *end example*]

Attributes	Description

val (Text Box Form Field Type Values)	Specifies the type of the text box form field, as defined by the simple type referenced below.
	
	[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:
	
	`<w:ffData>`
	`  <w:textInput>`
	`  <w:type w:val="currentDate" />`
	`  </w:textInput>`
	`</w:ffData>`
	
	The val attribute value of `currentDate` specifies that the contents of this form field should be the current date when the field is updated. *end example*]
	
	The possible values for this attribute are defined by the ST_FFTextType simple type (§17.18.28).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FFTextType) is located in §A.1. *end note*]

17.17 Miscellaneous Topics

This section covers topics not covered elsewhere within the WordprocessingML documentation.

17.17.1 Subdocuments

Within a WordprocessingML document, it is sometimes necessary to break a large document into two or more separate WordprocessingML document files, allowing each of these files to be distributed, edited, and handled independently.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: A book might consist of five chapters, each edited by a separate author. The editor for the book would therefore desire to create six WordprocessingML documents - one for each author to work on their chapter, and a main document which collates the content of the five chapters appropriately. *end example*]

When a WordprocessingML document is comprised of other WordprocessingML documents in this way, the resulting documents are called a master document and its subdocuments.

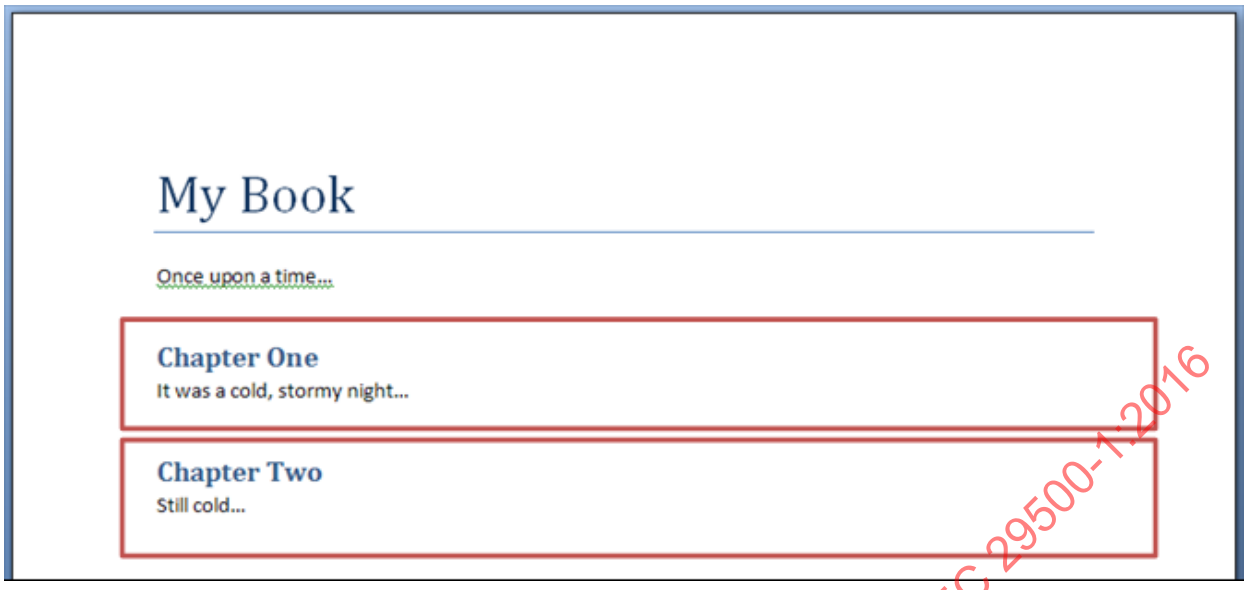
- A *subdocument* is a WordprocessingML document - there is no specific information in a document which classifies it as such, other than that it is incorporated into another document.
- A *master document* is a document which incorporates one or more subdocuments (as well as optional WordprocessingML content) to create a larger document

[*Example*: Consider a WordprocessingML document which is being used to write a book:

To allow this document to be written by multiple authors, each chapter in the book is placed in a separate file (the sections highlighted in red below):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



The result is three WordprocessingML documents:

- A master document (containing the title of the book, the first paragraph, and references to the subdocuments for each chapter)
- Two subdocuments (one for each chapter)

end example]

17.17.1.1 subDoc (Anchor for Subdocument Location)

This element specifies a location within a master document for the insertion of the contents of a specified subdocument. The specified subdocument's contents should appear at the specified location within the master document as needed, but shall remain part of the separate file specified by the subdocument location. The location of the subdocument shall be specified by the relationship whose Id attribute matches the id attribute on this element.

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/subDocument, is not present, or does not have a TargetMode attribute value of `External`, then the document shall be considered non-conformant.

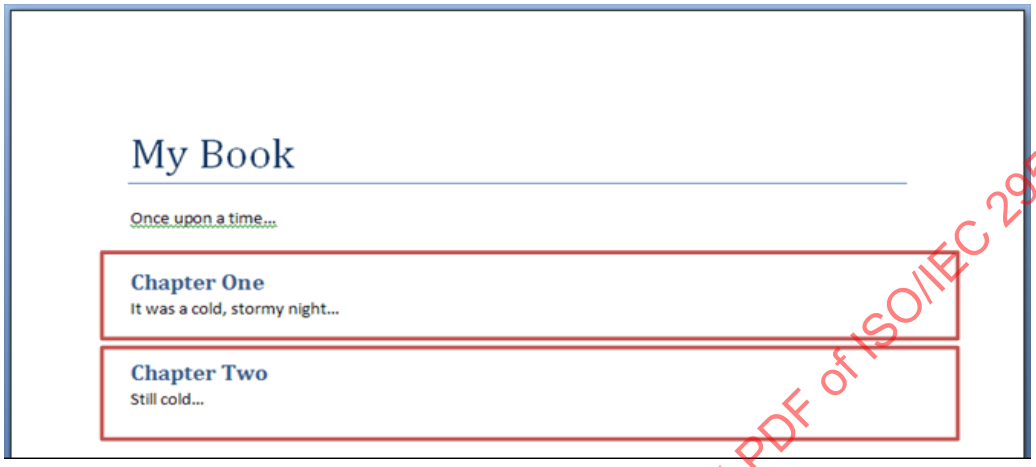
When a subdocument is displayed at the specified location within the master document, the following logic determines how the styles and formatting of the subdocument and associated master document shall be handled:

- All direct formatting in the subdocument is retained.
- For each style in each subdocument:
  - If a style with the same styleId attribute value exists in the master document, display the contents of the subdocument using the style in the master document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- Otherwise, import the style into the master document and display the content using the original style.
- If a subsequent subdocument also contains a style with the same styleId attribute value, the latter's style is ignored and the content is displayed using the version of the style imported into the master document.

[*Example*: Consider a book consisting of three chapters, two of which have been divided into subdocuments as follows (the red rectangle indicates the bounds of each subdocument's contents):



The resulting master document would consist of its own WordprocessingML content as well as subdocument anchors in the appropriate locations:

```
<w:body>
  <w:p>
    …
    <w:r>
      <w:t>My Book</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:r>
      <w:t>Once upon a time…</w:t>
    </w:r>
  </w:p>
  <w:p>
    <w:subDoc r:id="subDocRel1" />
  </w:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:p>
      <w:subDoc r:id="subDocRel2" />
    </w:p>
    <w:sectPr>
      …
    </w:sectPr>
  </w:body>
```

The two subDoc elements specify that the subdocuments targeted by the relationships with an ID of subDocRel1 and subDocRel2 must be imported in that order after the content of the first two paragraphs of content. Examining the contents of the corresponding relationship part item, we can see the targets for those relationships:

```
<Relationships … >
  …
  <Relationship Id="subDocRel1" TargetMode="External"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/subDocument"
Target="Chapter1.docx" />
  <Relationship Id="subDocRel2" TargetMode="External"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/subDocument"
Target="Chapter2.docx" />
  …
</Relationships>
```

The corresponding relationship part item shows that the two files to be imported are located in the same location as the current file and name `Chapter1.docx` and `Chapter2.docx` respectively. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/relationships/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.1. *end note*]

17.17.2 Alternative Format Import

When generating WordprocessingML documents, it is sometimes necessary to include existing document content (henceforth called *external content*) within the document. External content in a document is typically included because it was stored in a format other than the WordprocessingML format defined by ISO/IEC 29500.

As described in §11.3.1, conforming producers shall not create WordprocessingML packages which utilize this mechanism.

In order to facilitate the inclusion of such content without requiring its conversion as a prerequisite to its inclusion in a document, WordprocessingML includes the facility for applications to implement the import of external content in any format as part of a WordprocessingML document. This functionality, called external content import, allows the inclusion of content of an arbitrary content type within the WordprocessingML package, which shall then be opened and merged into the main document when the package is consumed by applications which understand that content type.

[*Example*: Consider a WordprocessingML document which is being created based on the following existing HTML content:

```
<html … >
  <body style="margin-left:200px;margin-top:50px">
    <p>Paragraph one.</p>
    <blockquote style="border:5px solid #00FFFF">Paragraph in a
blockquote.</blockquote>
    <p>Paragraph two.</p>
  </body>
</html>
```

This content could be converted to its WordprocessingML equivalents using the XML syntax defined by ISO/IEC 29500, or a more basic tool can use the external content import to include the HTML document within a WordprocessingML package, allowing a subsequent consumer of that content to import the resulting HTML. When the resulting WordprocessingML package is opened, the HTML document must be read (if it is an

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

alternate format understood by the consuming application) and migrated into the appropriate location in the main WordprocessingML document. *end example*]

17.17.2.1 altChunk (Anchor for Imported External Content)

This element specifies a location within a document for the insertion of the contents of a specified file containing external content to be imported into the main WordprocessingML document. The specified file's contents should appear at the specified location within the document, and can henceforth be emitted as regular WordprocessingML without distinction to its origin. The location of the external content to be imported shall be specified by the relationship whose Id attribute matches the id attribute on this element.

If the relationship type of the relationship specified by this element is not http://purl.oclc.org/ooxml/officeDocument/relationships/aFChunk, is not present, or does not have a TargetMode attribute value of `Internal`, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it should ignore the specified alternate content but continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[*Example*: Consider a WordprocessingML document consisting of contents which must be imported from the following HTML document:

```
<html … >
  <body style="margin-left:200px;margin-top:50px">
    <p>Paragraph one.</p>
    <blockquote style="border:5px solid #00FFFF">Paragraph in a
blockquote.</blockquote>
    <p>Paragraph two.</p>
  </body>
</html>
```

The resulting WordprocessingML host document would consist of its own WordprocessingML content as well as an external content import anchor in the appropriate location:

```
<w:body>
  <w:altChunk r:id="altChunk1" />
  <w:p/>
  <w:sectPr>
    …
  </w:sectPr>
</w:body>
```

The altChunk element specifies that the external content targeted by the relationship with an ID of `altChunk1` must be imported at the beginning of the document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<Relationships … >
  …
  <Relationship Id="altChunk1" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/aFChunk" Target="import.htm"
/>
  …
</Relationships>
```

The corresponding relationship part item shows that the file to be imported is located next to the main document and is named `import.htm`. *end example*]

Attributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part containing alternate content for import. If the specified relationship does not match the relationship type required by the parent element, then this document shall be considered to be non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId10" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*] If this attribute is omitted, the parent element shall be ignored. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AltChunk) is located in §A.1. *end note*]

17.17.2.2 altChunkPr (External Content Import Properties)

This element specifies the set of properties which shall be applied to the import of the external content specified by the parent altChunk element. Within ISO/IEC 29500, only one property is specified.

[*Example*: Consider a WordprocessingML document consisting of contents which contains an external content import anchor in the appropriate location:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:body>
  <w:altChunk r:id="altChunk1">
    <w:altChunkPr>
      <w:matchSrc w:val="false" />
    </w:altChunkPr>
  </w:altChunk>
  <w:p/>
  <w:sectPr>
    …
  </w:sectPr>
</w:body>
```

The altChunkPr element specifies the set of properties applied to the external content import when importing the specified content. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AltChunkPr) is located in §A.1. *end note*]

17.17.2.3 matchSrc (Keep Source Formatting on Import)

This element specifies if any style definitions present in the imported content shall be overridden by identical styles present in the host WordprocessingML document. If this element's val attribute is true, then any style exists in both the imported content and main document shall be maintained on the imported content by redefining the style name and/or ID as needed. Conversely, if this element's val attribute is false, any style which exists in both the imported content and main document shall apply the style form the main document in place of the style in the imported content.

If this element is omitted, then styles from the main document shall override identical styles from the imported content.

[*Example*: Consider a WordprocessingML document consisting of contents which contains an external content import anchor in the appropriate location:

```
<w:body>
  <w:altChunk r:id="altChunk1">
    <w:altChunkPr>
      <w:matchSrc w:val="true" />
    </w:altChunkPr>
  </w:altChunk>
  <w:p/>
  <w:sectPr>
    …
  </w:sectPr>
</w:body>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The matchSrc element has a val attribute value of `true`, which specifies that conflicting styles must be maintained when importing the specified content. For example, if the `Heading 1` style was defined in both places, then applications must ensure that the resulting document does not lose either instance of its formatting as appropriate. *end example*]

This element's content model is defined by the common boolean property definition in §17.17.4.

17.17.3 Roundtripping Alternate Content

WordprocessingML does not define a set of locations where applications should, whenever possible, attempt to store and roundtrip all non-taken choices in alternate content blocks. This behavior is therefore application-defined. For further discussion of alternate content blocks see §L.1.18.4.

17.17.4 Boolean Property (CT_OnOff)

This common complex type specifies a boolean attribute used throughout WordprocessingML.

Attributes	Description
val (On/Off Value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property shall be explicitly turned off. [*Example*: For example, consider the following on/off property: `<… w:val="false"/>` The val attribute explicitly declares that the property is `false`. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this complex type's content model (CT_OnOff) is located in §A.1. *end note*]

17.18 Simple Types

This is the complete list of simple types dedicated to WordprocessingML.

17.18.1 ST_AnnotationVMerge (Table Cell Vertical Merge Revision Type)

This simple type specifies the possible values for the vertical merge setting which applied to a table cell by a cell merge (or split) revision.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a two row by two column table in which the cells in the second column are merged, and this change is tracked as a revision. The annotation on the last cell in the table would appear as follows:

```
<w:tc>
  <w:tcPr>
    <w:cellMerge … w:vMerge="cont" />
  </w:tcPr>
  …
</w:tc>
```

The vmerge attribute value of cont specifies that the revision on the table cell resulted in it being merged with the previous set of vertically merged cells above it (whether that was one cell or many). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cont (Vertically Merged Cell)	Specifies that the revision resulted in this cell being vertically merged with the cell above it.
rest (Vertically Split Cell)	Specifies that the revision resulted in this cell being vertically split from the one above it.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnnotationVMerge) is located in §A.1. *end note*]

17.18.2 ST_Border (Border Styles)

This simple type specifies the kinds of borders which can be specified for WordprocessingML objects which have a border.

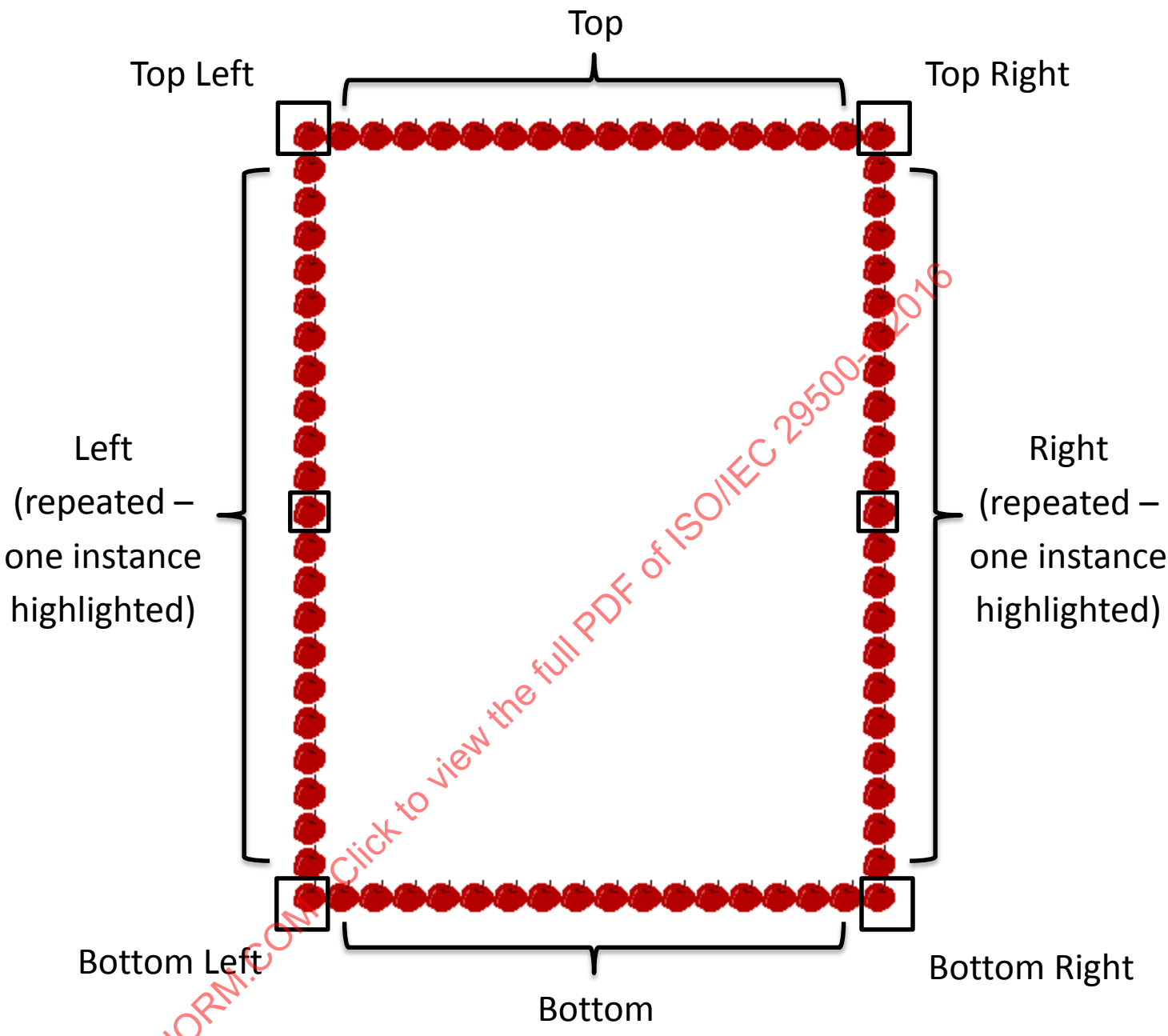
Borders can be separated into two types:

- *Line borders*, which specify a pattern to be used when drawing a line around the specified object.
- *Art borders*, which specify a repeated image to be used when drawing a border around the specified object.

Line borders can be specified on any object which allows a border, however, art borders can only be used as a border at the page level - the borders under the pgBorders element (§17.6.10).

For art borders, each border definition is specified by the combination of eight images, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

For the edge borders (top, bottom, left, and right), the border image is repeated as necessary to span the distance between the two intersecting corners. Electronic normative definitions of each piece of border art are included in Annex F.

[*Example*: Consider a left border resulting in the following WordprocessingML:

```
<w:left w:val="single" …/>
```

This border's val is `single`, indicating that the border style is a single line border. *end example*]

©ISO/IEC 2016 – All rights reserved

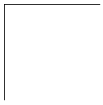
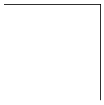
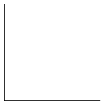
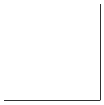
This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
apples (Apples Art Border)	Specifies an art border using the following images: All Sides:
archedScallops (Arched Scallops Art Border)	Specifies an art border using the following images: All Sides:
babyPacifier (Baby Pacifier Art Border)	Specifies an art border using the following images: All Sides:
babyRattle (Baby Rattle Art Border)	Specifies an art border using the following images: All Sides:
balloons3Colors (Three Color Balloons Art Border)	Specifies an art border using the following images: All Sides:
balloonsHotAir (Hot Air Balloons Art Border)	Specifies an art border using the following images: All Sides:
basicBlackDashes (Black Dash Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right:

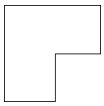
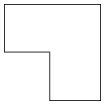
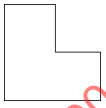
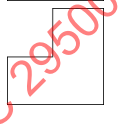
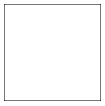
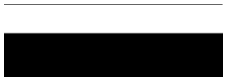
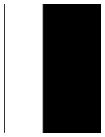
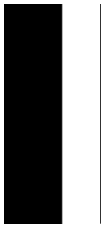
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Left: Top Right: Bottom Left: Bottom Right:
basicBlackDots (Black Dot Art Border)	Specifies an art border using the following images: All Sides:
basicBlackSquares (Black Square Art Border)	Specifies an art border using the following images: All Sides:
basicThinLines (Thin Line Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left:  Top Right:  Bottom Left:  Bottom Right: 
basicWhiteDashes (White Dash Art Border)	Specifies an art border using the following images: Top and Bottom:

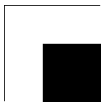
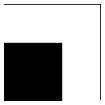
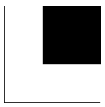
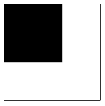
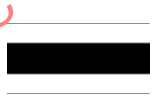
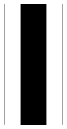
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Left and Right: Top Left:  Top Right:  Bottom Left:  Bottom Right: 
basicWhiteDots (White Dot Art Border)	Specifies an art border using the following images: All Sides:
basicWhiteSquares (White Square Art Border)	Specifies an art border using the following images: All Sides: 
basicWideInline (Wide Inline Art Border)	Specifies an art border using the following images: Top:  Bottom: Left:  Right: 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Left:  Top Right:  Bottom Left:  Bottom Right: 
basicWideMidline (Wide Midline Art Border)	Specifies an art border using the following images: Top and Bottom:  Left and Right:  Top Left: Top Right: Bottom Left: Bottom Right:
basicWideOutline (Wide Outline Art Border)	Specifies an art border using the following images: Top: Bottom: Left:

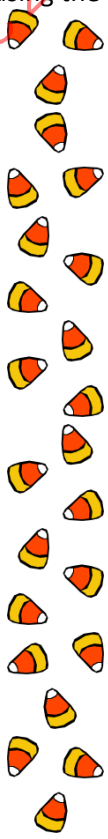
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Right: Top Left: Top Right: Bottom Left: Bottom Right:
bats (Bats Art Border)	Specifies an art border using the following images: All Sides:
birds (Birds Art Border)	Specifies an art border using the following images: Top, Bottom, Top Left, Top Right, Bottom Left and Bottom Right: Left and Right: [*Note*: Second image is blank. *end note*]
birdsFlight (Birds Flying Art Border)	Specifies an art border using the following images: All Sides:
cabins (Cabin Art Border)	Specifies an art border using the following images:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

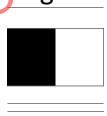
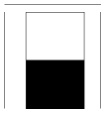
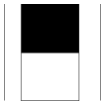
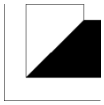
Enumeration Value	Description
	All Sides:
cakeSlice (Cake Art Border)	Specifies an art border using the following images: All Sides:
candyCorn (Candy Corn Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right: 
celticKnotwork (Knot Work Art Border)	Specifies an art border using the following images: Top: Bottom:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

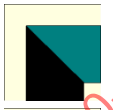
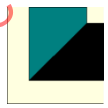
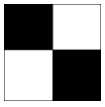
Enumeration Value	Description
	Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
certificateBanner (Certificate Banner Art Border)	Specifies an art border using the following images: Top: Bottom: Left: [Note: Image is blank. end note] Right: [Note: Image is blank. end note] Top Left: Top Right: Bottom Left: Bottom Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
chainLink (Chain Link Art Border)	Specifies an art border using the following images: All Sides:
champagneBottle (Champagne Bottle Art Border)	Specifies an art border using the following images: All Sides:
checkedBarBlack (Black and White Bar Art Border)	Specifies an art border using the following images: Top:  Bottom: Left:  Right:  Top Left: Top Right: Bottom Left:  Bottom Right:
checkedBarColor (Color Checked Bar Art Border)	Specifies an art border using the following images: Top:

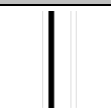
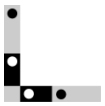
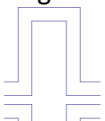
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom: Left: Right: Top Left:  Top Right: Bottom Left:  Bottom Right:
checkered (Checkerboard Art Border)	Specifies an art border using the following images: All Sides: 
christmasTree (Christmas Tree Art Border)	Specifies an art border using the following images: All Sides:
circlesLines (Circles And Lines Art Border)	Specifies an art border using the following images: Top: Bottom: Left:

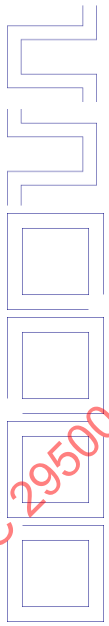
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Right:  Top Left: Top Right: Bottom Left: Bottom Right:
circlesRectangles (Circles and Rectangles Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left:  Bottom Right:
classicalWave (Wave Art Border)	Specifies an art border using the following images: Top:  Bottom:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Left: Right: Top Left: Top Right: Bottom Left: Bottom Right: 
clocks (Clocks Art Border)	Specifies an art border using the following images: All Sides:
compass (Compass Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left, Top Right, Bottom Left, and Bottom RIght:
confetti (Confetti Art Border)	Specifies an art border using the following images: Top: Bottom: Left:

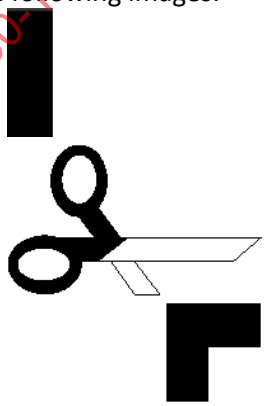
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Right: Top Left: Top Right: Bottom Left: Bottom Right:
confettiGrays (Confetti Art Border)	Specifies an art border using the following images: All Sides:
confettiOutline (Confetti Art Border)	Specifies an art border using the following images: All Sides:
confettiStreamers (Confetti Streamers Art Border)	Specifies an art border using the following images: All Sides:
confettiWhite (Confetti Art Border)	Specifies an art border using the following images: All Sides:
cornerTriangles (Corner Triangle Art Border)	Specifies an art border using the following images: Top, Bottom, Left, and Right: Top Left:

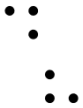
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Right: Bottom Left: Bottom Right:
couponCutoutDashes (Dashed Line Art Border)	Specifies an art border using the following images: Top, Bottom, Left, and Right:  Top Left: Top Right: Bottom Left: Bottom Right:
couponCutoutDots (Dotted Line Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Right:  Bottom Left: Bottom Right:
crazyMaze (Maze Art Border)	Specifies an art border using the following images: Top and Bottom: Left and RIght: Top Left, Top Right, Bottom Left and Bottom Right:
creaturesButterfly (Butterfly Art Border)	Specifies an art border using the following images: All Sides:
creaturesFish (Fish Art Border)	Specifies an art border using the following images: All Sides:
creaturesInsects (Insects Art Border)	Specifies an art border using the following images: Top: Bottom: Left and Right: Top Left and Top Right:

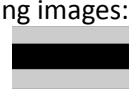
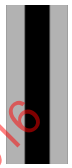
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left and Bottom Right:
creaturesLadyBug (Ladybug Art Border)	Specifies an art border using the following images: All Sides:
crossStitch (Cross-stitch Art Border)	Specifies an art border using the following images: All Sides:
cup (Cupid Art Border)	Specifies an art border using the following images: All Sides:
custom (Custom Defined Art Border)	Specifies a custom art border using the parent element's attributes to reference one or more customer art border images.
dashDotStroked (Dash Dot Strokes Line Border)	Specifies a line border consisting of a line with a series of alternating thin and thick strokes around the parent object. [*Example*: *end example*]
dashed (Dashed Line Border)	Specifies a line border consisting of a dashed line around the parent object. [*Example*: - - - - - - - - - - - - - - - *end example*]
dashSmallGap (Dashed Line Border)	Specifies a line border consisting of a dashed line with small gaps around the parent object. [*Example*: ----------

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	end example]
decoArch (Archway Art Border)	Specifies an art border using the following images: Top and Bottom:  Left and Right:  Top Left and Top Right: Bottom Left and Bottom Right:
decoArchColor (Color Archway Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left: Top Right: Bottom Left and Bottom Right:
decoBlocks (Blocks Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right:

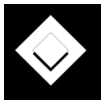
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Left: Top Right: Bottom Left: Bottom Right:
diamondsGray (Gray Diamond Art Border)	Specifies an art border using the following images: All Sides:
dotDash (Dot Dash Line Border)	Specifies a line border consisting of a alternating dotted and dashed line around the parent object. [*Example*: - - - - - - - - - - - - - - - - *end example*]
dotDotDash (Dot Dot Dash Line Border)	Specifies a line border consisting of a alternating dotted, dotted, dashed line around the parent object. [*Example*: - *end example*]
dotted (Dotted Line Border)	Specifies a line border consisting of a dotted line around the parent object. [*Example*: ..... *end example*]
double (Double Line Border)	Specifies a line border consisting of a double line around the parent object.

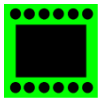
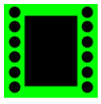
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: *end example*]
doubleD (Double D Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
doubleDiamonds (Diamond Art Border)	Specifies an art border using the following images: All Sides: 
doubleWave (Double Wave Line Border)	Specifies a line border consisting of a double wavy line around the parent object. [*Example*: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
earth1 (Earth Art Border)	Specifies an art border using the following images: All Sides:
earth2 (Earth Art Border)	Specifies an art border using the following images: All Sides:
earth3 (Earth Art Border)	Specifies an art border using the following images: All sides:
eclipsingSquares1 (Shadowed Square Art Border)	Specifies an art border using the following images: All Sides:
eclipsingSquares2 (Shadowed Square Art Border)	Specifies an art border using the following images: All Sides:
eggsBlack (Painted Egg Art Border)	Specifies an art border using the following images: All Sides:
fans (Fans Art Border)	Specifies an art border using the following images: All Sides:
film (Film Reel Art Border)	Specifies an art border using the following images: Top and Bottom:  Left and Right:  Top Left and Bottom Left:

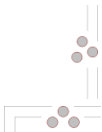
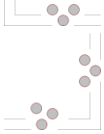
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Right and Bottom Right:
firecrackers (Firecracker Art Border)	Specifies an art border using the following images: All Sides:
flowersBlockPrint (Flowers Art Border)	Specifies an art border using the following images: All Sides:
flowersDaisies (Daisy Art Border)	Specifies an art border using the following images: All Sides:
flowersModern1 (Flowers Art Border)	Specifies an art border using the following images: All Sides:
flowersModern2 (Flowers Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left: Top Right: Bottom Left: Bottom Right:

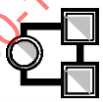
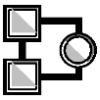
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
flowersPansy (Pansy Art Border)	Specifies an art border using the following images: All Sides:
flowersRedRose (Red Rose Art Border)	Specifies an art border using the following images: All Sides:
flowersRoses (Roses Art Border)	Specifies an art border using the following images: All Sides:
flowersTeacup (Flowers in a Teacup Art Border)	Specifies an art border using the following images: All Sides:
flowersTiny (Small Flower Art Border)	Specifies an art border using the following images: Top: Bottom: Left:  Right: Top Left: Top Right: Bottom Left:  Bottom Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
gems (Gems Art Border)	Specifies an art border using the following images: Top, Bottom, Left and Right Top Left: Top Right: Bottom Left:  Bottom Right: 
gingerbreadMan (Gingerbread Man Art Border)	Specifies an art border using the following images: All Sides:
gradient (Triangle Gradient Art Border)	Specifies an art border using the following images: All Sides:
handmade1 (Handmade Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Right: Bottom Left: Bottom Right:
handmade2 (Handmade Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
heartBalloon (Heart-Shaped Balloon Art Border)	Specifies an art border using the following images: All Sides:
heartGray (Gray Heart Art Border)	Specifies an art border using the following images:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	All Sides:
hearts (Hearts Art Border)	Specifies an art border using the following images: All Sides:
heebieJeebies (Pattern Art Border)	Specifies an art border using the following images: All Sides:
holly (Holly Art Border)	Specifies an art border using the following images: All Sides:
houseFunky (House Art Border)	Specifies an art border using the following images: All Sides:
hypnotic (Circular Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
iceCreamCones (Ice Cream Cone Art Border)	Specifies an art border using the following images: All Sides:
inset (Inset Line Border)	Specifies a line border consisting of an inset set of lines around the parent object. [*Example*: *end example*]
lightBulb (Light Bulb Art Border)	Specifies an art border using the following images: All Sides:
lightning1 (Lightning Art Border)	Specifies an art border using the following images: All Sides:
lightning2 (Lightning Art Border)	Specifies an art border using the following images: Top: Bottom: Left and Top Left: Right and Top RIght:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
mapleLeaf (Maple Leaf Art Border)	Specifies an art border using the following images: All Sides:
mapleMuffins (Muffin Art Border)	Specifies an art border using the following images: All Sides:
mapPins (Map Pins Art Border)	Specifies an art border using the following images: All Sides:
marquee (Marquee Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left, Top Right, Bottom Left, and Bottom Right:
marqueeToothed (Marquee Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left:

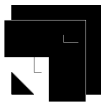
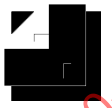
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Right:
moons (Moon Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
mosaic (Mosaic Art Border)	Specifies an art border using the following images: All Sides:
musicNotes (Musical Note Art Border)	Specifies an art border using the following images: All Sides:
nil (No Border)	Specifies that no border shall be applied to the current item.
none (No Border)	Specifies that no border shall be applied to the current item.
northwest (Patterned Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Left: Top Right:  Bottom Left: Bottom Right: 
outset (Outset Line Border)	Specifies a line border consisting of an outset set of lines around the parent object. [*Example*: *end example*]
ovals (Oval Art Border)	Specifies an art border using the following images: Top and Bottom: Right and Left: Top Left, Top Right, Bottom Left and Bottom Right:
packages (Package Art Border)	Specifies an art border using the following images: All Sides:
palmsBlack (Black Palm Tree Art Border)	Specifies an art border using the following images: All Sides:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
palmsColor (Color Palm Tree Art Border)	Specifies an art border using the following images: All Sides:
paperClips (Paper Clip Art Border)	Specifies an art border using the following images: All Sides:
papyrus (Papyrus Art Border)	Specifies an art using the following images: Top and Bottom: Right and Left: Top Left and Top Right: Bottom Left and Bottom Right:
partyFavor (Party Favor Art Border)	Specifies an art border using the following images: All Sides:
partyGlass (Party Glass Art Border)	Specifies an art border using the following images: All Sides:
pencils (Pencils Art Border)	Specifies an art border using the following images: All Sides:
people (Character Art Border)	Specifies an art border using the following images: All Sides:
peopleHats (Character With Hat Art Border)	Specifies an art border using the following images:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	All Sides:
peopleWaving (Waving Character Border)	Specifies an art border using the following images: All Sides:
poinsettias (Poinsettia Art Border)	Specifies an art border using the following images: All Sides:
postageStamp (Postage Stamp Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
pumpkin1 (Pumpkin Art Border)	Specifies an art border using the following images: All Sides:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
pushPinNote1 (Push Pin Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
pushPinNote2 (Push Pin Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Right:
pyramids (Pyramid Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left, Top Right, Bottom Left, and Bottom RIght:
pyramidsAbove (Pyramid Art Border)	Specifies an art border using the following images: All Sides:
quadrants (Quadrants Art Border)	Specifies an art border using the following images: Top and Bottom: Left and RIght: Top Left: Top Right: Bottom Left: Bottom Right:
rings (Rings Art Border)	Specifies an art border using the following images: All Sides:
safari (Safari Art Border)	Specifies an art border using the following images:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	All Sides:
sawtooth (Saw tooth Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
sawtoothGray (Gray Saw tooth Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
scaredCat (Scared Cat Art Border)	Specifies an art border using the following images: All Sides:
seattle (Umbrella Art Border)	Specifies an art border using the following images: All Sides:
shadowedSquares (Shadowed Squares Art Border)	Specifies an art border using the following images: All Sides:
shapes1 (Black and White Shapes Art Border)	Specifies an art border using the following images: Top and Bottom: Right and Left: Top Left, Top Right, Bottom Left and Bottom Right:
shapes2 (Black and White Art Border Two)	Specifies an art border using the following images: Top, Bottom, Left and Right: Top Left, Top Right, Bottom Left and Bottom Right:
sharksTeeth (Shark Tooth Art Border)	Specifies an art border using the following images:

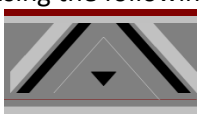
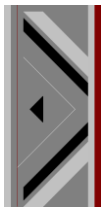
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	All Sides:
shorebirdTracks (Bird Tracks Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left, Top Right, Bottom Left and Bottom Right:
single (Single Line Border)	Specifies a line border consisting of a single line around the parent object. [Example: end example]
skyrocket (Rocket Art Border)	Specifies an art border using the following images: All Sides:
snowflakeFancy (Snowflake Art Border)	Specifies an art border using the following images: All Sides:
snowflakes (Snowflake Art Border)	Specifies an art border using the following images: All Sides:
sombrero (Sombrero Art Border)	Specifies an art border using the following images: All Sides:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
southwest (Southwest-themed Art Border)	Specifies an art border using the following images: Top:  Bottom: Left: Right:  Top Left: Top Right: Bottom Left: Bottom Right:
stars (Stars Art Border)	Specifies an art border using the following images: All Sides:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
stars3d (3-D Stars Art Border)	Specifies an art border using the following images: All Sides:
starsBlack (Stars Art Border)	Specifies an art border using the following images: All Sides:
starsShadowed (Stars With Shadows Art Border)	Specifies an art border using the following images: All Sides:
starsTop (Stars On Top Art Border)	Specifies an art border using the following images: Top, Top Left, and Top Right: Bottom, Bottom Left, and Bottom Right: Left and Right:
sun (Sun Art Border)	Specifies an art border using the following images: All Sides:
swirligig (Whirligig Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right:

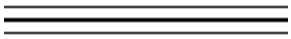
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
thick (Single Line Border)	Specifies a line border consisting of a single line around the parent object. [*Example*: ——————— *end example*]
thickThinLargeGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a large sized intermediate gap around the parent object. [*Example*: ═══════ *end example*]
thickThinMediumGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a medium sized intermediate gap around the parent object. [*Example*: ═══════ *end example*]
thickThinSmallGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a small intermediate gap around the parent object. [*Example*: ═══════ *end example*]
thinThickLargeGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	thin with a large sized intermediate gap between each around the parent object. [*Example*: *end example*]
thinThickMediumGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a medium sized intermediate gap between each around the parent object. [*Example*: *end example*]
thinThickSmallGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a small intermediate gap between each around the parent object. [*Example*: *end example*]
thinThickThinLargeGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a large sized intermediate gap around the parent object. [*Example*:  *end example*]
thinThickThinMediumGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a medium sized intermediate gap around the parent object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: *end example*]
thinThickThinSmallGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a small intermediate gap around the parent object. [*Example*: *end example*]
threeDEmboss (3D Embossed Line Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker towards the object. [*Example*: *end example*]
threeDEngrave (3D Engraved Line Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker away from the object. [*Example*: *end example*]
tornPaper (Torn Paper Art Border)	Specifies an art border using the following images: Top, Top Left and Top Right Bottom, Bottom Left and Bottom RIght: Left and RIght: [*Note*: Image is blank. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
tornPaperBlack (Black Torn Paper Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
trees (Tree Art Border)	Specifies an art border using the following images: All Sides:
triangle1 (Triangle Art Border One)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left, Top Right, Bottom Left, and Bottom Right:
triangle2 (Triangle Art Border Two)	Specifies an art border using the following images:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

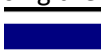
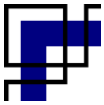
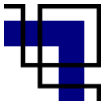
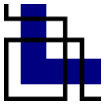
Enumeration Value	Description
	Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
triangleCircle1 (Triangle and Circle Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

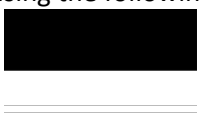
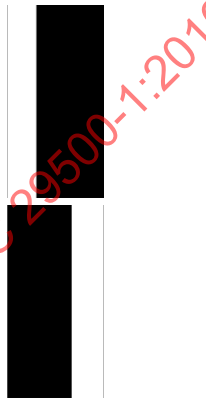
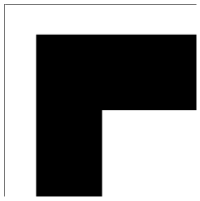
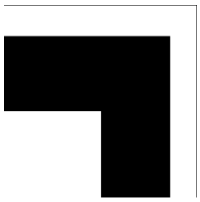
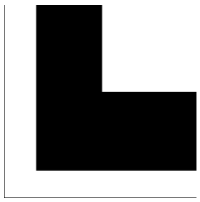
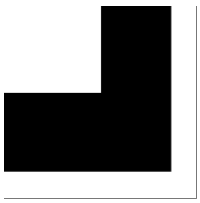
Enumeration Value	Description
	Bottom Right:
triangleCircle2 (Triangle and Circle Art Border Two)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
triangleParty (Triangle Art Border)	Specifies an art border using the following images: All Sides:
triangles (Triangles Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	Top Left: Top Right: Bottom Left: Bottom Right:
triple (Triple Line Border)	Specifies a line border consisting of a triple line around the parent object. [*Example*: *end example*]
twistedLines1 (Twisted Lines Art Border)	Specifies an art border using the following images: Top:  Bottom: Left: Right: Top Left:  Top Right:  Bottom Left:  Bottom Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
twistedLines2 (Twisted Lines Art Border)	Specifies an art border using the following images: Top:  Bottom: Left:  Right:  Top Left:  Top Right:  Bottom Left:  Bottom Right:
vine (Vine Art Border)	Specifies an art border using the following images:

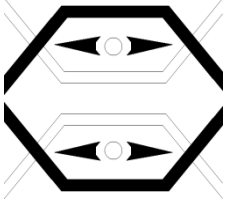
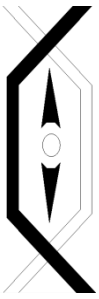
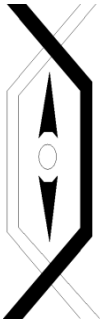
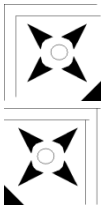
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

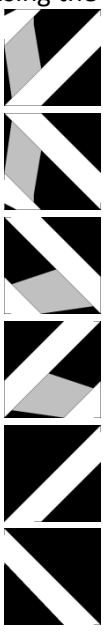
Enumeration Value	Description
	Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
wave (Wavy Line Border)	Specifies a line border consisting of a wavy line around the parent object. [*Example*: *end example*]
waveline (Wavy Line Art Border)	Specifies an art border using the following images: Top: Bottom: Left:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

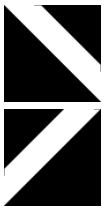
Enumeration Value	Description
	Right: Top Left: Top Right: Bottom Left: Bottom Right:
weavingAngles (Weaving Angles Art Border)	Specifies an art border using the following images: Top:  Bottom:  Left:  Right:  Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	Bottom Left: Bottom Right:
weavingBraid (Weaving Braid Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left, Top Right, Bottom Left, and Botton RIght:
weavingRibbon (Weaving Ribbon Art Border)	Specifies an art border using the following images: Top:  Bottom: Left: Right: Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left:  Bottom Right:
weavingStrips (Weaving Strips Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
whiteFlowers (White Flowers Art Border)	Specifies an art border using the following images: Top and Bottom: Left: Right: Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
woodwork (Woodwork Art Border)	Specifies an art border using the following images: Top and Bottom: Left and Right: Top Left, Top Right, Bottom Left, and Botton RIght:
xIllusions (Crisscross Art Border)	Specifies an art border using the following images: All Sides:
zanyTriangles (Triangle Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Bottom Left: Bottom Right:
zigZag (Zigzag Art Border)	Specifies an art border using the following images: Top: Bottom: Left: Right: Top Left: Top Right: Bottom Left: Bottom Right:
zigZagStitch (Zigzag stitch)	Specifies an art border using the following images: Top: Bottom: Left: Right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Top Left: Top Right: Bottom Left: Bottom Right:

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Border) is located in §A.1. *end note*]

17.18.3 ST_BrClear (Line Break Text Wrapping Restart Location)

This simple type specifies the set of possible restart locations which can be used as to determine the next available line when a break's type attribute has a value of `textWrapping`. This property only affects the restart location when the current run is being displayed on a line which does not span the full text extents due to the presence of a floating object (see enumeration values for details).

[*Example*: Consider a text wrapping break character which should force the restart location to the next line which spans the full width of the text extents of the page (there are no floating objects which interrupt the line).

This line break is of style `textWrapping`, since it must only advance to the next line, but the clear value must specify that this restart location must ignore all lines which are not of the full line width by specifying a value of `all`, as follows:

```
<w:br w:type="textWrapping" w:clear="all" />
```

This break must therefore not use the next available line, but rather the next available line ignoring all lines which do not span the full text width. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (Restart On Next Full Line)	Specifies that the text wrapping break shall advance the text to the next line in the WordprocessingML document which spans the full width of the line (i.e. the next line which is not interrupted by any floating

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	objects when those objects are positioned on the page at display time. [*Note*: This setting is typically used to place a single line of text next to a floating object for use as a caption. *end note*]

left (Restart In Next Text Region Unblocked on the Left)	Specifies that the text wrapping break shall behave as follows when this line intersects a floating object:
	
	If the parent paragraph is left-to-right:
	• If this is the leftmost region of text flow currently on this line,
	  • If a floating object occurs to the left of the break, advance the text to the next available line that does not have a floating object on the left.
	  • Otherwise, advance the text to the next position on the line where text can be displayed
	• Otherwise, treat this as a text wrapping break of type none.
	
	If the parent paragraph is right to left:
	If the object occurs to the left of the break, advance the text to the next available line that does not have a floating object on the left.
	• Otherwise, treat this as a text wrapping break of type none.
	
	In either case, if this line does not intersect a floating object, then treat this break as a text wrapping break of type none.

none (Restart On Next Line)	Specifies that the text wrapping break shall advance the text to the next line in the WordprocessingML document, regardless of its position left to right or the presence of any floating objects which intersect with the line, This is the setting for a typical line break in a document.
right (Restart In Next Text Region Unblocked on the Right)	Specifies that the text wrapping break shall behave as follows when this line intersects a floating object:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description

	If the parent paragraph is left-to-right:
	• If the object occurs to the right of the break, advance the text to the next available line that does not have a floating object on the right.
	• Otherwise, treat this as a text wrapping break of type none.
	If the parent paragraph is right to left:
	• If this is the rightmost region of text flow currently on this line,
	  • If a floating object occurs to the right of the break, advance the text to the next available line that does not have a floating object on the right.
	  • Otherwise, advance the text to the next position on the line where text can be displayed
	• Otherwise, treat this as a text wrapping break of type none.
	
	In either case, if this line does not intersect a floating object, then treat this break as a text wrapping break of type none.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BrClear) is located in §A.1. *end note*]

17.18.4 ST_BrType (Break Types)

This simple type specifies the possible kinds of break characters in a WordprocessingML document. The break type determines the next location where text shall be placed after this manual break is applied to the text contents (see enumeration values for details).

[*Example*: Consider a manual break which must advance the text to the next text column in the document, rather than just the next available line. This break would therefore be specified as follows:

```
<w:br w:type="column"/>
```

The type attribute specifies a value of `column`, which means that the break must force the next character in the document to be restarted on the next line in a new text column in the document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
column (Column Break)	Specifies that the current break shall restart itself on the next column available on the current page. If the current section is not divided into columns, or the column break occurs in the last column on the current page when displayed, then the restart location for text shall be the next page in the document.
page (Page Break)	Specifies that the current break shall restart itself on the next page of the document. Page breaks shall be ignored when present in frames.
textWrapping (Line Break)	Specifies that the current break shall restart itself on the next line in the document The determination of the next line shall be done subject to the value of the clear attribute on the specified break character.

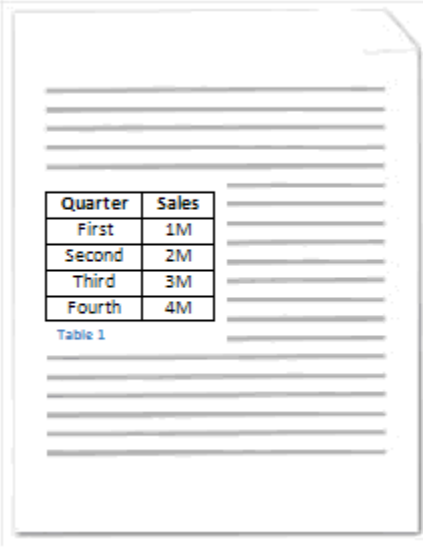
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BrType) is located in §A.1. *end note*]

17.18.5 ST_CaptionPos (Automatic Caption Positioning Values)

This simple type specifies the possible values can be used for the position of an automatically inserted caption on an object within this document. These values specify the position a given caption shall be take relative to the object it is used to label.

[*Example*: Consider a WordprocessingML document which should have all automatically inserted captions placed below the objects they are captioning, for example:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



This requirement is specified using the following WordprocessingML in the document settings:

```
<w:captions>
  <w:caption w:name="Table" w:pos="below" w:numFmt="decimal" />
</w:captions>
```

The pos attribute has a value of below, specifying that the caption must be placed below the newly inserted objects. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
above (Position Caption Above Object)	Specifies that an automatically inserted caption shall be positioned above the object that it is used to label.
below (Position Caption Below Object)	Specifies that an automatically inserted caption shall be positioned below the object that it is used to label.
left (Position Caption Left Of Object)	Specifies that an automatically inserted caption shall be positioned to the left of the object that it is used to label (the position where text typed immediately before the object would appear).
right (Position Caption Right Of Object)	Specifies that an automatically inserted caption shall be positioned to the right of the object that it is used to label (the position where text typed immediately after the object would appear).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CaptionPos) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.18.6 ST_ChapterSep (Chapter Separator Types)

This simple type specifies the character which shall be used to separate the chapter number from the page number for page numbers in a given section, when chapter numbers are being displayed.

[*Example*: Consider a section in a document in which the chapter must be separated from the page number using a colon character. This constraint would be specified using the following WordprocessingML:

```
<w:pgNumType w:chapSep="colon" w:chapStyle="1" />
```

The chapSep attribute declares that the chapter and page number must be separated by a colon (e.g. 1:1 for chapter one, page one). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
colon (Colon Chapter Separator)	Specifies that a colon character shall be used to separate the chapter number from the page number when page numbers are displayed. [Example: 1:1 for page one, section one. end example]
emDash (Em Dash Chapter Separator)	Specifies that an em dash character shall be used to separate the chapter number from the page number when page numbers are displayed. [*Example*: 1—1 for page one, section one. *end example*]
enDash (En Dash Chapter Separator)	Specifies that an en dash character shall be used to separate the chapter number from the page number when page numbers are displayed. [*Example*: 1–1 for page one, section one. *end example*]
hyphen (Hyphen Chapter Separator)	Specifies that a non-breaking hyphen character shall be used to separate the chapter number from the page number when page numbers are displayed. [*Example*: 1-1 for page one, section one. *end example*]
period (Period Chapter Separator)	Specifies that a period character shall be used to separate the chapter number from the page number when page numbers are displayed. [*Example*: 1.1 for page one, section one. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChapterSep) is located in §A.1. *end note*]

17.18.7 ST_CharacterSpacing (Character-Level Whitespace Compression Settings)

This simple type specifies the possible ways in which full-width characters in the current WordprocessingML document can be compressed to remove additional whitespace when the contents of this document are displayed, specifically by specifying the set(s) of characters which can be compressed to remove additional whitespace.

[*Example*: Consider the WordprocessingML below:

```
<w:characterSpacingControl w:val="doNotCompress" />
```

The characterSpacingControl element has a val attribute value of `doNotCompress`, which specifies that no character compression shall be applied to any character when the document is displayed. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
compressPunctuation (Compress Whitespace From Punctuation Characters)	Specifies that only whitespace characters shall have whitespace compression applied to them.
compressPunctuationAndJapaneseKana (Compress Whitespace From Both Japanese Kana And Punctuation Characters)	Specifies that whitespace and Japanese kana characters shall have whitespace compression applied to them.
doNotCompress (Do Not Compress Whitespace)	Specifies that characters shall not have whitespace compression applied to them.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CharacterSpacing) is located in §A.1. *end note*]

17.18.8 ST_CombineBrackets (Two Lines in One Enclosing Character Type)

This simple type specifies the kind of bracket character which shall be used to enclose the two lines in one text within the current run when displayed

[*Example*: Consider a paragraph with the text `two lines in one`, which must be displayed within a single logical line in the document and enclosed in curly brackets. This constraint would be specified as follows in the WordprocessingML:

```
<w:r>
  <w:rPr>
    <w:eastAsianLayout w:id="1" w:combine="on" w:combineBrackets="curly"/>
  </w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:t>two lines in one</w:t>
  </w:r>
```

The resulting text would be displayed on two sub lines within the other text on this line and enclosed within curly brackets when displayed. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
angle (Angle Brackets)	Specifies that angle bracket characters shall be used to enclose the contents of the current run's two lines in one text. [*Example*: <…> *end example*]
curly (Curly Brackets)	Specifies that curly bracket characters shall be used to enclose the contents of the current run's two lines in one text. [*Example*: {…} *end example*]
none (No Enclosing Brackets)	Specifies that no characters shall be used to enclose the contents of the current run's two lines in one text.
round (Round Brackets)	Specifies that round bracket characters shall be used to enclose the contents of the current run's two lines in one text. [*Example*: (…) *end example*]
square (Square Brackets)	Specifies that square bracket characters shall be used to enclose the contents of the current run's two lines in one text. [*Example*: […] *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CombineBrackets) is located in §A.1. *end note*]

17.18.9 ST_DateTime (Standard Date and Time Storage Format)

This simple type specifies that its contents contain a date in the standard XML Schema xsd:dateTime format, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:date w:fullDate="01-01-2006T12:00:00Z">
```

©ISO/IEC 2016 – All rights reserved

```
  …
</w:date>
```

In this case, the date in the realDate attribute is the full date associated with the parent date picker structured document. In every case, the value of this simple type is interpreted in the context of the parent element or attribute. *end example*]

This simple type's contents are a restriction of the W3C XML Schema dateTime datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DateTime) is located in §A.1. *end note*]

17.18.10 ST_DecimalNumber (Decimal Number Value)

This simple type specifies that its contents contain a whole decimal number (positive or negative), whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:pPr>
  <w:divId w:val="1512645511" />
</w:pPr>
```

The value of the val attribute is the ID of the associated HTML div.

However, consider the following fragment:

```
<w:ilvl w:val="1">
  …
</w:ilvl>
```

In this case, the decimal number in the val attribute is the ID of the associated numbering level. In each case, the value is interpreted in the context of the parent element. *end example*]

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DecimalNumber) is located in §A.1. *end note*]

17.18.11 ST_DecimalNumberOrPercent (Percentage Measurement)

This simple type specifies that its contents will contain a percentage-based value. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DecimalNumberOrPercent) is located in §A.1. *end note*]

17.18.12 ST_Direction (Bidirectional Direction Types)

This simple type specifies the possible values for bidirectional settings within a WordprocessingML document.

[*Example*: Consider the following bidirectional setting; in this case, a bidirectional override:

```
<w:bdo w:val="rtl">
  …
</w:bdo>
```

The val attribute explicitly declares that the override is applied right to left via the val attribute value of `rtl`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

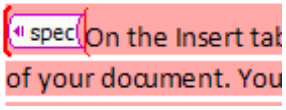
Enumeration Value	Description
ltr (Left toRight)	Specifies a left-to-right direction for the property defined by the parent XML element.
rtl (Right to Left)	Specifies a right-to-left direction for the property defined by the parent XML element.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Direction) is located in §A.1. *end note*]

17.18.13 ST_DisplacedByCustomXml (Location of Custom XML Markup Displacing an Annotation)

This simple type specifies the possible values for the location of a single custom XML element's start and/or end tag relative to the location of an annotation tag in document order. This enumeration shall be used to specify that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document.

[*Example*: Consider a paragraph with block level custom XML markup and two comment anchor annotations (one before and one after the custom XML element's physical representation), as follows:



Since all three of these items are around the entire paragraph, they are stored outside of the paragraph. However, in order to ensure that their relative positions are stored correctly, any annotation which must be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

displaced by the physical custom XML element specifies this information, resulting in the following WordprocessingML:

```
…
<w:commentRangeStart w:id="0" />
<w:commentRangeStart w:id="1" w:displacedByCustomXml="next" />
<w:customXml w:element="spec" … >
  <w:p>
    …
  </w:p>
</w:customXml>
…
```

The displacedByCustomXml attribute specifies that even though all three of these items are around the paragraph and is moved inside the paragraph to be represented physically, the comment with ID 0 must be inside the custom XML, but the comment with ID 1 must be displaced to stay outside of the relative location of the next custom XML element (the spec element). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
next (Displaced by Next Custom XML Markup Tag)	Specifies that this annotation anchor shall be displaced by the physical representation of the next element of custom XML markup in the document. If no custom XML markup exists in the same paragraph and after this anchor, then this setting shall be ignored.
prev (Displaced by Previous Custom XML Markup Tag)	Specifies that this annotation anchor shall be displaced by the physical representation of the previous element of custom XML markup in the document. If no custom XML markup exists in the same paragraph and directly before this anchor, then this setting shall be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DisplacedByCustomXml) is located in §A.1. *end note*]

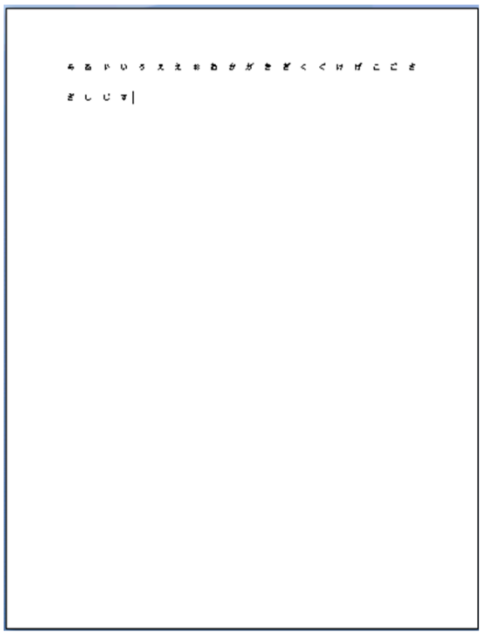
17.18.14 ST_DocGrid (Document Grid Types)

Specifies the kind of the current document grid, which defines the grid behavior.

 ©ISO/IEC 2016 – All rights reserved

The grid can define a grid which snaps all East Asian characters to grid positions, but leaves Latin text with its default spacing; a grid which adds the specified character pitch to all characters on each row; or a grid which affects only the line pitch for the current section.

[*Example*: Consider the document discussed above with the document grid defined to allow 20 characters per line, and 20 lines per page by snapping characters to the grid as follows:



This document has a type attribute of type ST_DocGrid and value snapToChars, which specifies that the grid must force East Asian characters to fit 20 to a line. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (No Document Grid)	Specifies that no document grid shall be applied to the contents of the current section in the document.
lines (Line Grid Only)	Specifies that the parent section shall have additional line pitch added to each line within it (as specified on the docGrid element (§17.6.5)) in order to maintain the specified number of lines per page.
linesAndChars (Line and Character Grid)	Specifies that the parent section shall have both the additional line pitch and character pitch added to each line and character within it (as specified on the docGrid element (§17.6.5)) in order to maintain a specific number of lines per page and characters per line.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	When this value is set, the input specified via the user interface can be allowed in exact number of line/character pitch units.
snapToChars (Character Grid Only)	Specifies that the parent section shall have both the additional line pitch and character pitch added to each line and character within it (as specified on the docGrid element (§17.6.5)) in order to maintain a specific number of lines per page and characters per line. When this value is set, the input specified via the user interface can be restricted to the number of lines per page and characters per line, with the consumer or producer translating this information based on the current font data to get the resulting line and character pitch values

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocGrid) is located in §A.1. *end note*]

17.18.15 ST_DocPartBehavior (Insertion Behavior Types)

This simple type specifies the possible sets of behaviors which can be applied to the contents of a single glossary document entry (§17.12.5) when it is added to the main document story of a WordprocessingML document.

[*Example*: Consider the WordprocessingML fragment for a glossary document entry containing a single run, defined as follows:

```
<w:docPart>
  <w:docPartPr>
    <w:behaviors>
      <w:behavior w:val="p"/>
    </w:behaviors>
    …
  </w:docPartPr>
  <w:docPartBody>
    <w:p>
      <w:r>
        <w:t>Sample entry.</w:t>
      </w:r>
    </w:p>
  </w:docPartBody>
</w:docPart>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The behavior element of type ST_DocPartBehavior has a value of p, which specifies that the contents of the parent glossary document entry must be inserted in their own paragraph when they are added to the contents of a document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
content (Insert Content At Specified Location)	Specifies that when the glossary document entry is inserted into the main document contents of the document, it shall be inserted normally as defined above. This includes ensuring that the final paragraph which is included in the part is not inserted, and its run content is added to the paragraph into which the current part is being inserted.
p (Ensure Entry Is In New Paragraph)	Specifies that the glossary document entry shall be added into its own unique paragraph, by failing to remove the last paragraph from the entry's contents when they are added to the document.
pg (Ensure Entry Is On New Page)	Specifies that the glossary document entry shall be added into its own new page, by preceding the entry with a blank paragraph whose only content is a page break character.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocPartBehavior) is located in §A.1. *end note*]

17.18.16 ST_DocPartGallery (Entry Gallery Types)

This simple type specifies possible settings for the predefined gallery into which a glossary document part shall be classified. This classification, although its enumeration values can be interpreted to imply semantics around the contents of the parent glossary document entry, shall only be used to classify and sort this entry (via an application or a user interface).

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:docPartPr>
  <w:category>
    <w:name w:val="Internal Memo Covers" />
    <w:gallery w:val="coverPg" />
  </w:category>
  …
</w:docPartPr>
```

The gallery element with a value of coverPg specifies that the gallery categorization applied to the current entry, for the purposes of classification or user interface sorting, puts this entry into the Cover Pages classification. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
any (All Galleries)	Specifies that this glossary document entry shall be associated with all possible gallery classification values.
autoTxt (AutoText Gallery)	Specifies that this glossary document entry shall be associated with the AutoText gallery classification.
bib (Bibliography Gallery)	Specifies that this glossary document entry shall be associated with the Bibliography gallery classification.
coverPg (Cover Page Gallery)	Specifies that this glossary document entry shall be associated with the Cover Page gallery classification.
custAutoTxt (Custom AutoText Gallery)	Specifies that this glossary document entry shall be associated with the Custom AutoText gallery classification.
custBib (Custom Bibliography Gallery)	Specifies that this glossary document entry shall be associated with the Custom Bibliography gallery classification.
custCoverPg (Custom Cover Page Gallery)	Specifies that this glossary document entry shall be associated with the Custom Cover Page gallery classification.
custEq (Custom Equation Gallery)	Specifies that this glossary document entry shall be associated with the Custom Equation gallery classification.
custFtrs (Custom Footer Gallery)	Specifies that this glossary document entry shall be associated with the Custom Footer gallery classification.
custHdrs (Custom Header Gallery)	Specifies that this glossary document entry shall be associated with the Custom Header gallery classification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
custom1 (Custom 1 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 1 gallery classification.
custom2 (Custom 2 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 2 gallery classification.
custom3 (Custom 3 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 3 gallery classification.
custom4 (Custom 4 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 4 gallery classification.
custom5 (Custom 5 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 5 gallery classification.
custPgNum (Custom Page Number Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number gallery classification.
custPgNumB (Custom Page Number At Bottom Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Bottom gallery classification.
custPgNumMargins (Custom Page Number At Margins Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Margins gallery classification.
custPgNumT (Custom Page Number At Top Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Top gallery classification.
custQuickParts (Custom Quick Parts Gallery)	Specifies that this glossary document entry shall be associated with the Custom Quick Parts gallery classification.
custTblOfContents (Custom Table of Contents Gallery)	Specifies that this glossary document entry shall be associated with the Custom Table of Contents gallery classification.
custTbls (Custom Table Gallery)	Specifies that this glossary document entry shall be associated with the Custom Tables gallery classification.
custTxtBox (Custom Text Box Gallery)	Specifies that this glossary document entry shall be associated with the Custom Text Box gallery classification.
custWatermarks (Custom Watermark Gallery)	Specifies that this glossary document entry shall be associated with the Custom Watermark gallery classification.
default (No Gallery Classification)	Specifies that this glossary document entry shall not have a gallery classification.
docParts (Document Parts Gallery)	Specifies that this glossary document entry shall be associated with the Document Parts gallery

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	classification.
eq (Equations Gallery)	Specifies that this glossary document entry shall be associated with the Equations gallery classification.
ftrs (Footers Gallery)	Specifies that this glossary document entry shall be associated with the Footers gallery classification.
hdrs (Headers Gallery)	Specifies that this glossary document entry shall be associated with the Headers gallery classification.
pgNum (Page Numbers Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers gallery classification.
pgNumB (Page Numbers At Bottom Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Bottom gallery classification.
pgNumMargins (Page Numbers At Margins Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Margins gallery classification.
pgNumT (Page Numbers At Top Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Top gallery classification.
placeholder (Structured Document Tag Placeholder Text Gallery)	Specifies that this glossary document entry shall be associated with the Structured Document Tag Placeholder Text gallery classification.
tblOfContents (Table of Contents Gallery)	Specifies that this glossary document entry shall be associated with the Table of Contents gallery classification.
tbls (Table Gallery)	Specifies that this glossary document entry shall be associated with the Tables gallery classification.
txtBox (Text Box Gallery)	Specifies that this glossary document entry shall be associated with the Text Box gallery classification.
watermarks (Watermark Gallery)	Specifies that this glossary document entry shall be associated with the Watermark gallery classification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocPartGallery) is located in §A.1. *end note*]

17.18.17 ST_DocPartType (Entry Types)

This simple type specifies the possible entry types which can be applied to the properties of a single glossary document entry (§17.12.5). Each of these entry types can, based on their values, influence the visibility and behavior of the parent glossary document entry.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  <w:types>
    <w:type w:val="bbPlcHdr" />
  </w:types>
  …
</w:docPartPr>
```

The type element with a value of `bbPlcHdr` specifies that the parent glossary document entry must be treated as if it was the placeholder text for one or more structured document tags in the document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autoExp (Automatically Replace Name With Content)	Specifies that the type of the current glossary document entry shall allow the entry to be automatically inserted into the document whenever its name is entered into an application.
bbPlcHdr (Structured Document Tag Placeholder Text)	Specifies that the type of the current glossary document entry shall be structured document tag placeholder text.
formFld (Form Field Help Text)	Specifies that the type of the current glossary document entry shall be form field help text.
none (No Type)	Specifies no type information for the current glossary document entry.
normal (Normal)	Specifies that the type of the current glossary document entry shall be normal (i.e. a regular glossary document entry).
speller (AutoCorrect Entry)	Specifies that the type of the current glossary document entry shall be associated with the spelling and grammar tools.
toolbar (AutoText User Interface Entry)	Specifies that the type of the current glossary document entry shall be associated with a special grouping of entries associated with a single piece of user interface.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocPartType) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.18.18 ST_DocProtect (Document Protection Types)

This simple type specifies the possible set of editing restrictions which can be enforced on a given WordprocessingML document.

[*Example*: Consider a WordprocessingML document that contains the following WordprocessingML specifying that hosting applications must enforce read-only protection for a given document:

```
<w:documentProtection w:edit="readOnly" w:enforcement="1" />
```

The edit attribute has a value of `readOnly` and an enforcement attribute with a value of `1`, specifying that read-only document protection must be enforced on the given document. e*nd example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
comments (Allow Editing of Comments)	Specifies that the edits made to this document shall be restricted to: • The insertion and deletion of comments within the document • The editing of regions delimited by range permissions which match the editing rights of the user account which is performing the editing.
forms (Allow Editing of Form Fields)	Specifies that the edits made to this document shall be restricted to: • The editing of form fields in sections where the formProt element (§17.6.6) has a value of `true`. • No restrictions in sections where the formProt element has a value of `false`.
none (No Editing Restrictions)	Specifies that no editing restrictions have been applied to the document.
readOnly (Allow No Editing)	Specifies that the edits made to this document shall be restricted to: • The editing of regions delimited by range permissions which match the editing rights of the user account which is performing the editing.
trackedChanges (Allow Editing With Revision Tracking)	Specifies that the edits made to this document shall be tracked as revisions. This value shall imply the presence of the trackRevisions element (§17.15.1.89), and applications shall not allow that element's state to be changed to `false`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocProtect) is located in §A.1. *end note*]

17.18.19 ST_DocType (Document Classification Values)

This simple type specifies the possible classifications that can be used for a WordprocessingML document.

The following possible values for this type are reserved:

Value	Description
eMail (E-Mail Message)	Specifies that this document shall be classified as an e-mail message.
letter (Letter)	Specifies that this document shall be classified as a letter.
notSpecified (Default Document)	Specifies that this document shall be classified as a default document.

[*Example*: Consider a set of WordprocessingML documents which should be classified as 'letters'. This classification would be specified using the following WordprocessingML in the document settings of these documents:

```
<w:documentType w:val="letter" />
```

The documentType element's val attribute is equal to `letter`, specifying that the hosting application must apply the behaviors it has specified for letters to the given WordprocessingML document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DocType) is located in §A.1. *end note*]

17.18.20 ST_DropCap (Text Frame Drop Cap Location)

This simple type specifies the location which shall be used to position a drop cap text frame when the contents of that text frame are displayed in the anchor paragraph at display time.

[*Note*: Although a drop cap is simply a text frame, the values of this simple type are used to determine how the cap should be positioned relative to the following non-frame paragraph in relative terms (see enumeration values), rather than relying on absolute sizing. *end note*]

[*Example*: Consider the following paragraph containing a text frame which should be positioned as a drop cap:

```
<w:p>
  <w:pPr>
    <w:framePr w:dropCap="margin" w:lines="3" w:hSpace="432" w:wrap="around"
w:vAnchor="text" w:hAnchor="page" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </w:pPr>
    <w:r>
      <w:t>A</w:t>
    </w:r>
  </w:p>
```

The dropCap attribute specifies a value of `margin`, so this drop cap is placed outside of the text margin before the start of the current text. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
drop (Drop Cap Inside Margin)	Specifies that the drop cap text frame shall be positioned inside the text margin on the anchor paragraph when this text frame is displayed in the document.
margin (Drop Cap Outside Margin)	Specifies that the drop cap text frame shall be positioned outside of the text margin on the anchor paragraph when this text frame is displayed in the document.
none (Not Drop Cap)	Specifies that this text frame is not a drop cap text frame.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DropCap) is located in §A.1. *end note*]

17.18.21 ST_EdGrp (Range Permision Editing Group)

This simple type specifies the set of possible aliases (or editing groups) which can be used as aliases to determine if the current user shall be allowed to edit a single range defined by a range permission within a document. This mechanism simply provides a set of predefined editing groups which can be associated with user accounts by applications in any desired manner.

[*Example*: Consider a range permission defined as follows:

```
<w:permStart w:id="0" w:edGrp="editors" … />
…
<w:permEnd w:id="0" />
```

The edGrp attribute value of `editors` specifies that only user(s) who the current application associates with the editors group must be allowed to edit the contents between the start and end markers when document protection is being enforced. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
administrators (Administrator Group)	Specifies that users associated with the Administrators group shall be allowed to edit range permissions using this editing group when document protection is enabled.
contributors (Contributors Group)	Specifies that users associated with the Contributors group shall be allowed to edit range permissions using this editing group when document protection is enabled.
current (Current Group)	Specifies that users associated with the Current group shall be allowed to edit range permissions using this editing group when document protection is enabled.
editors (Editors Group)	Specifies that users associated with the Editors group shall be allowed to edit range permissions using this editing group when document protection is enabled.
everyone (All Users Have Editing Permissions)	Specifies that all users that open the document shall be allowed to edit range permissions using this editing group when document protection is enabled.
none (No Users Have Editing Permissions)	Specifies that none of the users that open the document shall be allowed to edit range permissions using this editing group when document protection is enabled.
owners (Owners Group)	Specifies that users associated with the Owners group shall be allowed to edit range permissions using this editing group when document protection is enabled.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EdGrp) is located in §A.1. *end note*]

17.18.22 ST_EdnPos (Endnote Positioning Location)

This simple type specifies the possible positions of endnotes in a document.

[*Example*: Consider a document in which endnotes must be positioned at the end of the section. The section properties for this section must be declared as follows:

```
<w:settings>
  <w:endnotePr>
    <w:pos w:val="sectEnd" />
  </w:endnotePr>
  …
</w:settings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The val attribute is sectEnd, therefore the position of endnotes is specified to be at the end of the section. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
docEnd (Endnotes Positioned at End of Document)	Specifies that all endnotes shall be placed at the end of the current document, regardless of which section they are referenced within.
sectEnd (Endnotes Positioned at End of Section)	Specifies that endnotes shall be placed at the end of the section in which they are referenced. An endnote which is never referenced is never displayed.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EdnPos) is located in §A.1. *end note*]

17.18.23 ST_EighthPointMeasure (Measurement in Eighths of a Point)

This simple type specifies that its contents contain a positive whole number, whose contents consist of a measurement in eighths of a point (equivalent to 1/576th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of 24 whose simple type is ST_EighthPointMeasure. This attribute value specifies a size in eighths of a point (24 eighths of a point = 3 points). *end example*]

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber datatype (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EighthPointMeasure) is located in §A.1. *end note*]

17.18.24 ST_Em (Emphasis Mark Type)

This simple type specifies an enumerated list of emphasis marks, any one of which may be selected to be applied to each non-space character in a run. When displayed, the position of the emphasis mark relative to the character to which it is applied is language- and writing-direction-dependent. When displayed, the glyph used for the emphasis mark is implementation-dependent.

[*Example*: Consider a run of text that is to have a dot emphasis mark applied to it. This is specified using the following WordprocessingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:rPr>
  <w:em w:val="dot"/>
</w:rPr>
```

end example]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
circle (Circle Emphasis Mark)	Specifies that the emphasis mark is a circle. [*Note*: The intent is to have an emphasis mark similar to the following: (written horizontally in Japanese) 中 (written vertically in Japanese) 中 *end note*]
comma (Comma Emphasis Mark)	Specifies that the emphasis mark is a comma. [*Note*: The intent is to have an emphasis mark similar to the following: (written horizontally in Japanese) き (written vertically in Japanese) き *end note*]
dot (Dot Emphasis Mark)	Specifies that the emphasis mark is a dot. [*Note*: The intent is to have an emphasis mark similar to the following: (written horizontally in Simplified or Traditional Chinese) 千 (written vertically in Simplified or Traditional Chinese) 千 (written horizontally in Japanese) き (written vertically in Japanese) き *end note*]
none (No Emphasis Mark)	Specifies that no emphasis mark shall be applied to any characters in the run.
underDot (Dot Emphasis Mark Below Characters)	Specifies that the emphasis mark is a dot that shall be rendered below each character in horizontal writing and on the left in vertical writing. [*Note*: The intent is to have an emphasis mark similar to the following: ..(written horizontally in Japanese)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	き ..(written vertically in Japanese) き *end note*] [*Note*: Ordinarily, the position of the emphasis mark relative to the character to which it is applied is determined automatically by the language- and writing-direction. As such, the position need not be specified explicitly, and the use of this value is discouraged. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Em) is located in §A.1. *end note*]

17.18.25 ST_FFHelpTextVal (Help Text Value)

This simple type specifies the format of optional help text which can be associated with the parent form field.

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:helpText w:type="text" w:val="Example help text."/>
      <w:ddList>
        …
      </w:ddList>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMDROPDOWN </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The helpText element specifies the help text for the parent form field - in this case, literal help text consisting of the string `Example help text.` *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 256 Unicode scalar values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FFHelpTextVal) is located in §A.1. *end note*]

17.18.26 ST_FFName (Form Field Name Value)

This simple type specifies the format of the name which can be associated with the parent form field.

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:name w:val=”FirstName” />
      <w:textInput>
        …
      </w:textInput>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMTEXT </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="separate"/>
</w:r>
<w:r>
  <w:t>1</w:t>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The name element specifies that the name of the current form field is `FirstName`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 65 Unicode scalar values.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FFName) is located in §A.1. *end note*]

17.18.27 ST_FFStatusTextVal (Status Text Value)

This simple type specifies the format of optional status text which can be associated with the parent form field.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following WordprocessingML fragment for a drop-down list form field:

```
<w:r>
  <w:fldChar w:fldCharType="begin">
    <w:ffData>
      <w:statusText w:type="text" w:val="Example status text."/>
      <w:ddList>
        …
      </w:ddList>
    </w:ffData>
  </w:fldChar>
</w:r>
<w:r>
  <w:instrText> FORMDROPDOWN </w:instrText>
</w:r>
<w:r>
  <w:fldChar w:fldCharType="end"/>
</w:r>
```

The statusText element specifies the status text for the parent form field - in this case, literal text consisting of the string `Example status text.` *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 140 characters.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FFStatusTextVal) is located in §A.1. *end note*]

17.18.28 ST_FFTextType (Text Box Form Field Type Values)

This simple type specifies the possible types of the contents of a text box form field.

[*Example*: Consider the following WordprocessingML fragment for the properties of a text box form field:

```
<w:ffData>
  <w:textInput>
    <w:type w:val="number" />
    <w:maxLength w:val="4" />
    <w:format w:val="0.00" />
  </w:textInput>
</w:ffData>
```

The type element specifies that the contents of this form field should be handled as a `number` by an application. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
calculated (Field Calculation)	Specifies that the contents of this text box form field shall be the result of the field calculation specified by the corresponding default element (§17.16.10). This field should not be directly editable when the editing of form fields is enabled.
currentDate (Current Date Display)	Specifies that the contents of this text box form field shall be the current date when the field is updated.
currentTime (Current Time Display)	Specifies that the contents of this text box form field shall be the current time when the field is updated.
date (Date)	Specifies that the contents of this text box form field shall be treated as a date.
number (Number)	Specifies that the contents of this text box form field shall be treated as a number value.
regular (Text Box)	Specifies that this text form field is a plain text field (no additional content restrictions).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FFTextType) is located in §A.1. *end note*]

17.18.29 ST_FldCharType (Complex Field Character Type)

This simple type specifies the possible values for the type of a single complex field character in the document.

[*Example*: Consider the following WordprocessingML for a complex field character:

```
…
<w:fldChar w:fldCharType="separate" />
…
```

The type attribute value of `separate` specifies that this is a complex field separator character; therefore it is being used to separate the field codes from the field contents in a complex field. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
begin (Start Character)	Specifies that the character is a start character, which defines the start of a complex field.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
end (End Character)	Specifies that the character is an end character, which defines the end of a complex field.
separate (Separator Character)	Specifies that the character is a separator character, which defines the end of the field codes and the start of the field result for a complex field.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FldCharType) is located in §A.1. *end note*]

17.18.30 ST_FontFamily (Font Family Value)

This simple type specifies possible values for the font family of a font.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Calibri">
  <w:family w:val="swiss" />
  …
</w:font>
```

The family element specifies via its val attribute value of swiss that this font is part of the Swiss family. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (No Font Family)	Specifies that information about a font's font family does not exist.
decorative (Novelty Font)	Specifies the Novelty font family.
modern (Monospace Font)	Specifies a monospace font with or without serifs (monospace fonts are usually modern).
roman (Proportional Font With Serifs)	Specifies a proportional font with serifs.
script (Script Font)	Specifies a script font designed to mimic the appearance of handwriting.
swiss (Proportional Font Without Serifs)	Specifies a proportional font without serifs.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontFamily) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

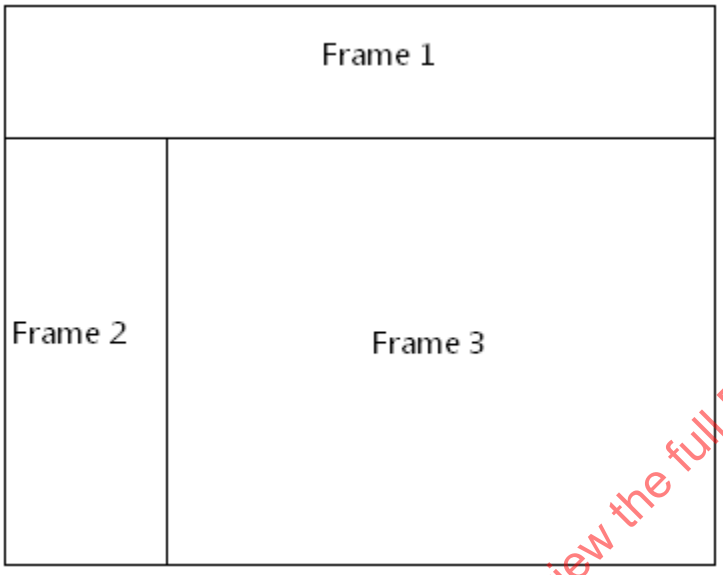
©ISO/IEC 2016 – All rights reserved

17.18.31 ST_FrameLayout (Frameset Layout Order)

This simple type specifies the possible order in which the frames (and nested framesets) in a frameset can be displayed. When a frameset is created, it can only contain frames which are stacked in one direction:

- Vertically (one on top of another)
- Horizontally (one next to another)

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  <w:frameLayout w:val="rows" />
  <w:frame>
    …
  </w:frame>
  <w:frameset>
    <w:frameLayout w:val="cols" />
    <w:frame>
      …
    </w:frame>
    <w:frame>
      …
    </w:frame>
  </w:frameset>
</w:frameset>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The frameLayout element specifies that the outer frameset is consists of the single frame and the child frameset stacked vertically, and an inner nested frameset consisting of two frames stacked horizontally. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

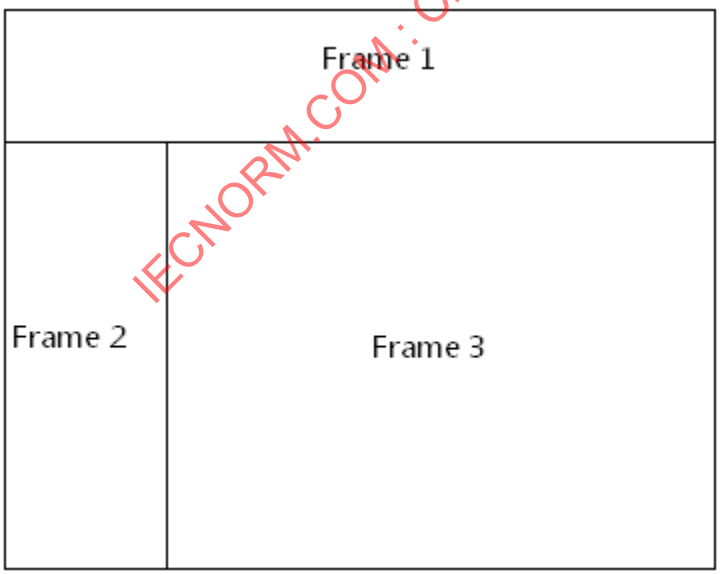
Enumeration Value	Description
cols (Stack Frames Horizontally)	Specifies that the frames in the frameset shall be stacked horizontally next to each other in left to right order.
none (Do Not Stack Frames)	Specifies that no frames shall be shown in the frameset.
rows (Stack Frames Vertically)	Specifies that the frames in the frameset shall be stacked vertically next to each other in top to bottom order.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FrameLayout) is located in §A.1. *end note*]

17.18.32 ST_FrameScrollbar (Frame Scrollbar Visibility)

This simple type specifies the possible settings for when a scrollbar shall be visible for the contents of the current frame.

[*Example*: Consider a WordprocessingML document which serves as the frameset container for a frameset consisting of the following three frames:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The frameset properties for this document are specified by the following WordprocessingML within the web page settings:

```
<w:frameset>
  …
  <w:frameset>
    …
    <w:frame>
      <w:name w:val="Frame 2" />
      <w:scrollbar w:val="auto" />
    </w:frame>
    …
  </w:frameset>
</w:frameset>
```

The scrollbar element has a val attribute of auto, which specifies that the frame must only display a scrollbar when it is needed to display all of its content. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatically Show Scrollbar As Needed)	Specifies that the scrollbar for a frame shall automatically be hidden and/or displayed as needed based on the length of the contents.
off (Never Show Scrollbar)	Specifies that the scrollbar for a frame shall always be hidden.
on (Always Show Scrollbar)	Specifies that the scrollbar for a frame shall always be displayed (even when not needed).

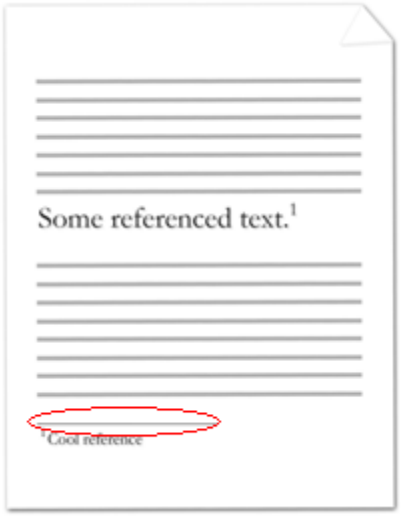
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FrameScrollbar) is located in §A.1. *end note*]

17.18.33 ST_FtnEdn (Footnote or Endnote Type)

This simple type specifies the possible types of footnotes and endnotes which can be specified in a WordprocessingML document.

[*Example*: Consider a document with a single footnote at the bottom of the first page. This footnote must be separated from the text by the separator footnote (the footnote explicitly used to separate text from the footnote list (circled in red below):

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



This footnote type would be declared as follows in the WordprocessingML:

```
<w:footnote w:type="separator" w:id="0">
  …
</w:footnote>
```

In this example, the footnote has an attribute value of separator, specifies when this footnote must be used. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
continuationNotice (Continuation Notice Separator)	Specifies that this footnote or endnote is a continuation notice footnote or endnote. *Continuation notice footnotes and endnotes* are used when the footnotes or endnotes exceed the length allowed on a single page. When this happens, this footnote or endnote shall be placed on the bottom of each page where the note shall continue to indicate that fact to the reader.
continuationSeparator (Continuation Separator)	Specifies that this footnote or endnote is a continuation separator footnote or endnote. *Continuation separator footnotes and endnotes* are used when the footnotes or endnotes exceed the length allowed on a single page. When this happens, this footnote or endnote shall be placed between the main text contents and the continued footnotes/endnotes on all subsequent pages of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	document.
normal (Normal Footnote/Endnote)	Specifies that this footnote or endnote is a normal footnote or endnote, and can be referenced by main document content.
separator (Separator)	Specifies that this footnote or endnote is a separator footnote or endnote. *Separator footnotes and endnotes* are used to indicate the separation between the main document's content and the footnotes or endnotes to indicate that fact to the reader.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FtnEdn) is located in §A.1. *end note*]

17.18.34 ST_FtnPos (Footnote Positioning Location)

This simple type specifies the position of footnotes in the document.

[*Example*: Consider a document in which footnotes must be positioned beneath their text. The footnote properties for this document must be declared as follows:

```
<w:sectPr>
  <w:footnotePr>
    <w:pos w:val="beneathText" />
  </w:footnotePr>
  …
</w:sectPr>
```

The val attribute is beneathText, therefore the position of footnotes is specified to be beneath the page's text. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
beneathText (Footnotes Positioned Beneath Text)	Specifies that footnotes shall be displayed immediately after the last line of text on the page on which the note reference mark appears.
docEnd (Footnotes Positioned At End of Document)	Specifies that all footnotes shall be placed at the end of the current document, regardless of which section they are referenced within.
pageBottom (Footnotes Positioned at Page Bottom)	Specifies that footnotes shall be displayed at the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	bottom margin of the page on which the note reference mark appears.
sectEnd (Footnotes Positioned At End of Section)	Specifies that all footnotes shall be placed at the end of the section in which they are referenced. A footnote which is never referenced is never displayed.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FtnPos) is located in §A.1. *end note*]

17.18.35 ST_HAnchor (Horizontal Anchor Location)

This simple type specifies the horizontal position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final horizontal position of the object in the document.

[*Example*: Consider a text frame which should be positioned one inch to the right of its column in a left-to-right document. This text frame would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:framePr … w:x="1440" w:hAnchor="margin" />
</w:pPr>
```

These frame horizontal anchor properties specify that they are relative to the anchor paragraph's `margin` (the text margin excluding any indents). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
margin (Relative To Margin)	Specifies that the parent object shall be horizontally anchored to the text margins. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the text margin.
page (Relative to Page)	Specifies that the parent object shall be horizontally anchored to the page edge. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
text (Relative to Text Extents)	Specifies that the parent object shall be horizontally anchored to the text extents. This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the text in the anchor paragraph (including text indentations on that paragraph within the text margins).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HAnchor) is located in §A.1. *end note*]

17.18.36 ST_HdrFtr (Header or Footer Type)

This simple type specifies the possible types of headers and footers which can be specified for a given header or footer reference in a document. This value determines the page(s) on which the current header or footer shall be displayed.

[*Example*: Consider a WordprocessingML section which specifies the following header reference:

```
<w:headerReference r:id="rId10" w:type="first" />
```

The resulting section must use the specified header part for the `first` page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (Default Header or Footer)	Specifies that this header or footer shall appear on every page in this section which is not overridden with a specific `even` or `first` page header/footer. In a section with all three types specified, this header/footer type shall be used on all odd numbered pages (counting from the first page in the section, not the section numbering).
even (Even Numbered Pages Only)	Specifies that this header or footer shall appear on all even numbered pages in this section (counting from the first page in the section, not the section numbering). The appearance of this header or footer is contingent on the setting of the evenAndOddHeaders element (§17.10.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
first (First Page Only)	Specifies that this header or footer shall appear on the first page in this section. The appearance of this header or footer is contingent on the setting of the titlePg element (§17.10.6).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HdrFtr) is located in §A.1. *end note*]

17.18.37 ST_HeightRule (Height Rule)

This simple type specifies the logic which shall be used to calculate the height of the parent object when it is displayed in the document.

[*Example*: Consider the following table row:

```
<w:trPr>
  <w:trHeight w:hRule="atLeast" w:val="2189" />
</w:trPr>
```

The val attribute specifies a value of 2189 twentieths of a point, so this table row is a minimum of 2189 twentieths of a point high regardless of its contents, since its hRule value is set to `atLeast`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
atLeast (Minimum Height)	Specifies that the height of the parent object shall be at least the value specified, but can be expanded to fit its content as needed.
auto (Determine Height Based On Contents)	Specifies that the height of the parent object shall be automatically determined by the size of its contents, with no predetermined minimum or maximum size.
exact (Exact Height)	Specifies that the height of the parent object shall be exactly the value specified, regardless of the size of the contents of the object. If the contents are too large for the specified height, then they shall be clipped.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HeightRule) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.18.38 ST_HexColor (Color Value)

This simple type specifies that its contents contain one of the following:

- A color values in RRGGBB format (ST_HexColorRGB)
- The enumeration value `auto` (ST_HexColorAuto)

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider a border color with value `auto`, as follows:

```
<w:bottom … w:color="auto"/>
```

This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*]

This simple type is a union of the following types:

- The ST_HexColorRGB simple type (§22.9.2.5).
- The ST_HexColorAuto simple type (§17.18.39).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HexColor) is located in §A.1. *end note*]

17.18.39 ST_HexColorAuto ('Automatic' Color Value)

This simple type specifies that its contents contain the enumeration value `auto`. This value shall be used to specify an automatically determined color value, the meaning of which is interpreted based on the context of the parent XML element.

[*Example*: Consider a border color with value `auto`, as follows:

```
<w:bottom … w:color="auto"/>
```

This color therefore can be automatically be modified by a consumer as appropriate, for example, in order to ensure that the border can be distinguished against the page's background color. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatically Determined Color)	Specifies that the color value can automatically be defined when this document is processed, based on the display context.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HexColorAuto) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.18.40 ST_HighlightColor (Text Highlight Colors)

This simple type specifies the possible values for highlighting colors which can be applied as a background behind the contents of a text run.

[*Example*: Consider a run within a paragraph which has yellow text highlighting using the highlight element. This formatting is specified using the following WordprocessingML:

```
<w:rPr>
  <w:highlight w:val="yellow" />
</w:rPr>
```

The resulting run would have `yellow` highlighting visible over its contents. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
black (Black Highlighting Color)	Specifies that the text highlighting color for this run shall be black. The hexadecimal RGB value for this setting shall be `000000`.
blue (Blue Highlighting Color)	Specifies that the text highlighting color for this run shall be blue. The hexadecimal RGB value for this setting shall be `0000FF`.
cyan (Cyan Highlighting Color)	Specifies that the text highlighting color for this run shall be cyan. The hexadecimal RGB value for this setting shall be `00FFFF`.
darkBlue (Dark Blue Highlighting Color)	Specifies that the text highlighting color for this run shall be dark blue. The hexadecimal RGB value for this setting shall be `00008B`.
darkCyan (Dark Cyan Highlighting Color)	Specifies that the text highlighting color for this run shall be dark cyan. The hexadecimal RGB value for this setting shall be `008B8B`.
darkGray (Dark Gray Highlighting Color)	Specifies that the text highlighting color for this run shall be dark gray.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	The hexadecimal RGB value for this setting shall be A9A9A9.
darkGreen (Dark Green Highlighting Color)	Specifies that the text highlighting color for this run shall be dark green. The hexadecimal RGB value for this setting shall be 006400.
darkMagenta (Dark Magenta Highlighting Color)	Specifies that the text highlighting color for this run shall be dark magenta. The hexadecimal RGB value for this setting shall be 800080.
darkRed (Dark Red Highlighting Color)	Specifies that the text highlighting color for this run shall be dark red. The hexadecimal RGB value for this setting shall be 8B0000.
darkYellow (Dark Yellow Highlighting Color)	Specifies that the text highlighting color for this run shall be dark cyan. The hexadecimal RGB value for this setting shall be 808000.
green (Green Highlighting Color)	Specifies that the text highlighting color for this run shall be green. The hexadecimal RGB value for this setting shall be 00FF00.
lightGray (Light Gray Highlighting Color)	Specifies that the text highlighting color for this run shall be light gray. The hexadecimal RGB value for this setting shall be D3D3D3.
magenta (Magenta Highlighting Color)	Specifies that the text highlighting color for this run shall be magenta. The hexadecimal RGB value for this setting shall be FF00FF.
none (No Text Highlighting)	Specifies that this text run shall have no text highlighting applied to its contents.
red (Red Highlighting Color)	Specifies that the text highlighting color for this run shall be red.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	The hexadecimal RGB value for this setting shall be FF0000.
white (White Highlighting Color)	Specifies that the text highlighting color for this run shall be white. The hexadecimal RGB value for this setting shall be FFFFFF.
yellow (Yellow Highlighting Color)	Specifies that the text highlighting color for this run shall be yellow. The hexadecimal RGB value for this setting shall be FFFF00.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HighlightColor) is located in §A.1. *end note*]

17.18.41 ST_Hint (Font Type Hint)

This simple type specifies information used to decide how to format any characters in the current run for which the font type is otherwise ambiguous.

Certain characters can be mapped into more than one of the font slot categories described in the parent element. This attribute shall be used to determine how ambiguities in this run shall be handled. [*Note*: This can be used to handle the formatting on the paragraph mark glyph, and other characters that are not stored as text in the WordprocessingML document. Some printable characters can be mapped to more than one font slot, such as Unicode glyph U+2026 'HORIZONTAL ELLIPSIS'. *end note*]

[*Example*: Consider the run representing the paragraph mark glyph, which is not stored as a physical character. Since this could therefore be formatted with any of the fonts specified for the run, this ambiguity is resolved using the following WordprocessingML:

```
<w:pPr>
  <w:rPr>
    <w:rFonts w:hint="eastAsia" />
  </w:rPr>
</w:pPr>
```

The hint attribute specifies that some characters in the run use an East Asian font. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cs (Complex Script Font)	Specifies that the font hint for this text run shall be to use the Complex Script font defined for the run via the style hierarchy.
default (no font hint)	Specifies that no hint shall apply to this text run.
eastAsia (East Asian Font)	Specifies that the font hint for this text run shall be to use the East Asian font defined for the run via the style hierarchy.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Hint) is located in §A.1. *end note*]

17.18.42 ST_HpsMeasure (Measurement in Half-Points)

This simple type specifies that its contents contain either:

- A positive whole number, whose contents consist of a measurement in half-points (equivalent to 1/144th of an inch), or
- A positive decimal number immediately followed by a unit identifier.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of 72 whose simple type is ST_HpsMeasure. This attribute value specifies a size of one-half of an inch or 36 points (72 halves of a point = 36 points = 0.5 inches). *end example*]

[*Example*: Consider an attribute value of 12.7mm whose type is ST_HpsMeasure. This attribute value specifies a size of 0.0127 meter or one-half of an inch or 36 points. *end example*]

This simple type is a union of the following types:

- The ST_PositiveUniversalMeasure simple type (§22.9.2.12).
- The ST_UnsignedDecimalNumber simple type (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HpsMeasure) is located in §A.1. *end note*]

17.18.43 ST_InfoTextType (Help or Status Text Type)

This simple type specifies the possible values for the type of help or status text which can be associated with a form field.

[*Example*: Consider the following WordprocessingML fragment for a form field:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:ffData>
  <w:helpText w:type="text" w:val="Example help text." />
</w:ffData>
```

The type attribute has a value of text, which specifies that the text in the val attribute is the literal help text for this form field. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autoText (Glossary Document Entry)	Specifies that the value specified by the parent XML element's val attribute shall be interpreted as the name of a glossary document entry whose contents contain the help or status text.
text (Literal Text)	Specifies that the value specified by the parent XML element's val attribute shall be interpreted as the literal text for the help or status text.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_InfoTextType) is located in §A.1. *end note*]

17.18.44 ST_Jc (Horizontal Alignment Type)

This simple type specifies all types of alignment which are available to be applied to objects in a WordprocessingML document.

[*Example*: Consider a paragraph which is aligned to the trailing edge of text flow. This requirement would be specified as follows in the WordprocessingML markup:

```
<w:pPr>
  <w:jc w:val="end" />
</w:pPr>
```

The val attribute's value of end specifies that the content must be right-aligned on the page for a left-to-right paragraph, and left justified for a right-to-left paragraph. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
both (Justified)	Specifies that the text shall be justified between both of the text margins in the document. The lowKashida setting shall also be applied to Arabic

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	text when this setting is applied. This type of justification shall only affect the inter-word spacing on each line, and not the inter-character spacing within each word when justifying its contents.
center (Align Center)	Specifies that the text shall be centered on the line between both of the text margins in the document.
distribute (Distribute All Characters Equally)	Specifies that the text shall be justified between both of the text margins in the document. This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to all characters on the line.
end (Align to Trailing Edge)	Specifies that the text shall be aligned on the trailing text margin in the document (right for left-to-right paragraphs; left for right-to-left paragraphs).
highKashida (Widest Kashida Length)	Specifies that the kashida length for text in the current paragraph shall be extended to its widest possible length. This setting only affects *kashidas*, which are special characters used to extend the joiner between two Arabic characters. [*Note*: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. *end note*] [*Example*: The following example illustrates each type of kashida: Low: هذا إمتيحان Medium: هذا إمتيحـــان High: هذا إمتيحـــــان *end example*]
lowKashida (Low Kashida Length)	Specifies that the kashida length for text in the current paragraph shall be extended to a slightly longer length. This setting shall also be applied to Arabic text when the both setting is applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	This setting only affects *kashidas*, which are special characters used to extend the joiner between two Arabic characters. [*Note*: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. *end note*] [*Example*: The following example illustrates each type of kashida: Low: هذا إمتيحان Medium: هذا إمتيحـــان High: هذا إمتيحـــــان *end example*]
mediumKashida (Medium Kashida Length)	Specifies that the kashida length for text in the current paragraph shall be extended to a medium length determined by the consumer. This setting only affects *kashidas*, which are special characters used to extend the joiner between two Arabic characters. [*Note*: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. *end note*] [*Example*: The following example illustrates each type of kashida: Low: هذا إمتيحان Medium: هذا إمتيحـــان High: هذا إمتيحـــــان *end example*]
numTab (Align to List Tab)	Specifies that the text shall be aligned to the list tab, which is the tab stop after the numbering for the current paragraph. If the current paragraph has no numbering, this setting has no effect.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: This justification style is used for backwards compatibility with earlier word processors, and should be avoided in favor of hanging paragraph indentation. *end note*]
start (Align To Leading Edge)	Specifies that the text shall be aligned on the leading text margin in the document (left for left-to-right paragraphs; right for right-to-left paragraphs).
thaiDistribute (Thai Language Justification)	Specifies that the text shall be justified with an optimization for Thai. This type of justification shall affect both the inter-word spacing on each line, and the inter-character spacing between each word when justifying its contents, unlike both justification. This difference is created in that the inter-character space is increased slightly in order to ensure that the additional space created by the justification is reduced. [*Note*: This setting is different from justification in that the reduction in inter-character spacing would be inappropriate in Western languages. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Jc) is located in §A.1. *end note*]

17.18.45 ST_JcTable (Table Alignment Type)

This simple type specifies all types of alignment that are available to be applied to tables in a WordprocessingML document.

[*Example*: Consider a table row that is right aligned. This requirement would be specified as follows in the WordprocessingML markup:

```
<w:trPr>
  <w:jc w:val="end" />
</w:trPr>
```

The val attribute's value of end specifies that the table is right-aligned on the page (assuming the table is aligned left-to-right. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Align Center)	Specifies that the table shall be centered on the line

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	between both of the text margins in the document.
end (Align to Trailing Edge)	Specifies that the table shall be aligned to the trailing edge of the text flow – the right text margin (for a left-to-right table); or the left text margin (for a right-to-left table) in the document. (See §17.4.1)
start (Align to Starting Edge)	Specifies that the table shall be aligned to the leading edge of the text flow – the left text margin (for a left-to-right table); or the right text margin (for a right-to-left table) in the document. (See §17.4.1)

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_JcTable) is located in §A.1. *end note*]

17.18.46 ST_LevelSuffix (Content Between Numbering Symbol and Paragraph Text)

This simple type specifies the types of content which shall be possible between a given numbering level's text and the text of every numbered paragraph which references that numbering level.

[*Example:* Consider the numbered paragraph below:

1. Test

In this example, a space exists between the numbering symbol `1.` and the numbered paragraph text `Test`. The space would be specified in WordprocessingML as follows:

```
<w:lvl w:ilvl="0">
  …
  <w:suff w:val="space" />
  …
</w:lvl>
```

The suff element with an attribute value of space specifies that the character between the numbering's level text and the paragraph text must be a space. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
nothing (Nothing Between Numbering and Text)	Specifies that no character shall be displayed between the numbering level's text and the contents of the paragraph when displaying the numbered paragraph.
space (Space Between Numbering and Text)	Specifies that a space character shall be displayed between the numbering level's text and the contents

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	of the paragraph when displaying the numbered paragraph.
tab (Tab Between Numbering and Text)	Specifies that a tab character shall be displayed between the numbering level's text and the contents of the paragraph when displaying the numbered paragraph. This tab shall follow normal tab stop rules to determine its length.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LevelSuffix) is located in §A.1. *end note*]

17.18.47 ST_LineNumberRestart (Line Numbering Restart Position)

This simple type specifies when the line numbering in the parent section shall be reset to its restart value. The line numbering increments for each line (even if the line number itself is not displayed) until it reaches the restart point specified by this element.

[*Example*: Consider the line numbering used on each page of this document, which specifies that line numbering must restart at the top of each new page. This line numbering setting would be defined using the following WordprocessingML:

```
<w:lnNumType w:restart="newPage" .../>
```

The restart attribute is of type ST_LineNumberRestart, and a value of newPage specifies that the line numbers must restart at the top of each page to the value specified by the start attribute. In this case, newPage is the default, so this value could have been omitted entirely. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
continuous (Continue Line Numbering From Previous Section)	Specifies that line numbering for the parent section shall continue from the line numbering from the end of the previous section, if any.
newPage (Restart Line Numbering on Each Page)	Specifies that line numbering for the parent section shall restart to the starting value whenever a new page is displayed.
newSection (Restart Line Numbering for Each Section)	Specifies that line numbering for the parent section shall restart to the starting value whenever the parent begins.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineNumberRestart) is located in §A.1. *end note*]

17.18.48 ST_LineSpacingRule (Line Spacing Rule)

This simple type specifies the logic which shall be used to calculate the line spacing of the parent object when it is displayed in the document.

[*Example*: Consider the following WordprocessingML paragraph:

```
<w:pPr>
  <w:spacing w:line="276" w:lineRule="auto" />
</w:pPr>
```

This paragraph specifies that the spacing in each line should be automatically calculated using 1.15 times (276 divided by 240) the normal single spacing calculation. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
atLeast (Minimum Line Height)	Specifies that the height of the line shall be at least the value specified, but might be expanded to fit its content as needed.
auto (Automatically Determined Line Height)	Specifies that the line spacing of the parent object shall be automatically determined by the size of its contents, with no predetermined minimum or maximum size.
exact (Exact Line Height)	Specifies that the height of the line shall be exactly the value specified, regardless of the size of the contents of the contents. If the contents are too large for the specified height, then they shall be clipped as necessary.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineSpacingRule) is located in §A.1. *end note*]

17.18.49 ST_Lock (Locking Types)

This simple type specifies the possible set of locking behaviors that can be applied to the contents of the nearest ancestor structured document tag when the contents of this document are edited by an application (whether through a user interface or directly).

[*Example*: Consider the following plain text structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    <w:lock w:val="sdtLocked"/>
    …
    <w:text/>
  </w:sdtPr>
  …
</w:sdt>
```

This plain text structured document tag's properties contain a lock element, specifying locking behaviors for the structured document tag. Since the locking val attribute value is sdtLocked, this locking setting must specify that the contents of the structured document tag can be edited, but the structured document tag itself must not be deleted from the document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
contentLocked (Contents Cannot Be Edited At Runtime)	Specifies that the editing restriction applied to the nearest ancestor structured document tag shall be as follows: • This structured document tag's contents shall not be editable • This structured document tag can be deleted in its entirety (but only entirely, no sub portion of it can be deleted)
sdtContentLocked (Contents Cannot Be Edited At Runtime And SDT Cannot Be Deleted)	Specifies that the editing restriction applied to the nearest ancestor structured document tag shall be as follows: • This structured document tag's contents shall not be editable • This structured document tag shall not be deleted in its entirety
sdtLocked (SDT Cannot Be Deleted)	Specifies that the editing restriction applied to the nearest ancestor structured document tag shall be as follows: • This structured document tag's contents shall be editable • This structured document tag shall not be deleted in its entirety
unlocked (No Locking)	Specifies that no special locking behaviors shall be applied to the nearest ancestor structured document tag. The default behaviors as specified on the lock element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	(§17.5.2.23) shall be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Lock) is located in §A.1. *end note*]

17.18.50 ST_LongHexNumber (Eight Digit Hexadecimal Value)

This simple type specifies a number value specified as a four octet (eight digit) hexadecimal number, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following value for a node of simple type ST_LongHexNumber: 00BE2C6C.

This value is permitted, as it contains four hexadecimal octets, each an encoding of an octet of the actual decimal number value. *end example*]

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 8 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LongHexNumber) is located in §A.1. *end note*]

17.18.51 ST_MacroName (Script Subroutine Name Value)

This simple type specifies a subroutine in a scripting language which can be executed based on the context of the parent XML element. The language and location of this subroutine can be determined using any method desired by an application.

[*Example*: Consider the following WordprocessingML fragment for the properties of a form field:

```
<w:ffData>
  <w:exitMacro w:val="TestExitFunction" />
</w:ffData>
```

The exitMacro element specifies that any application which processes this file should attempt to locate and execute a scripting subroutine called TestExitFunction when the contents of the form field are exited. If this subroutine cannot be located or executed, then this setting is silently ignored. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 33 Unicode scalar values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MacroName) is located in §A.1. *end note*]

17.18.52 ST_MailMergeDataType (Mail Merge Data Source Type Values)

This simple type specifies the data source access mechanism used to connect to the data source for a mail merge. This setting is purely a suggestion of the data source access mechanism which shall be used, and can be ignored in favor of an alternative mechanism if one is present.

Although this any access mechanism value can be specified, the following values are reserved:

Value	Description
native (Office Data Source Object Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source via the data stored in the Office Data Source Object (ODSO) interface (§17.14.25).
odbc (Open Database Connectivity Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source via ODBC.
query (Query Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source using an external query file.
soap (SOAP Data Source)	Specifies that a given WordprocessingML document has been connected to a data source using SOAP.
spreadsheet (Spreadsheet Data Source)	Specifies that a given WordprocessingML document has been connected to a spreadsheet.
textFile (Text File Data Source)	Specifies that a given WordprocessingML document has been connected to a text file.
xQuery (Text File Data Source)	Specifies that a given WordprocessingML document has been connected to an external data source via XQuery.
xmlFile (XML File Data Source)	Specifies that a given WordprocessingML document has been connected to an XML file.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document:

```
<w:dataType w:val="odbc" />
```

The dataType element's val attribute is equal to odbc, specifying that the given merged WordprocessingML document has been connected to an external data source via the ODBC. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MailMergeDataType) is located in §A.1. *end note*]

17.18.53 ST_MailMergeDest (Merged Document Destination Types)

This simple type specifies the possible results which can be generated when a mail merge is carried out on a given WordprocessingML source document. In other words, this element is used to specify what is to be done with the merged documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.

[*Example*: Consider a WordprocessingML source document containing the following WordprocessingML:

```
<w:mailMerge>
  <w:destination w:val="newDocument" />
  …
</w:mailMerge>
```

The destination element's val attribute is set to `newDocument`, specifying that when the mail merge is carried out, the source document must be used to generate a specified number of new documents, which can be handled as appropriate. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
email (Send Merged Documents as E-mail Messages)	Specifies that conforming hosting applications shall generate emails using the documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.
fax (Send Merged Documents as Faxes)	Specifies that conforming hosting applications shall generate faxes using the documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.
newDocument (Send Merged Documents to New Documents)	Specifies that conforming hosting applications shall generate new documents by populating the fields within a given merged WordprocessingML document with data from the specified external data source.
printer (Send Merged Documents to Printer)	Specifies that conforming hosting applications shall print the documents that result from populating the fields within a given merged WordprocessingML document with external data from the specified external data source.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MailMergeDest) is located in §A.1. *end note*]

17.18.54 ST_MailMergeDocType (Source Document Types)

This simple types specifies the possible types for a given WordprocessingML source document.

[*Example*: Consider the WordprocessingML below:

```
<w:mailMerge>
  <w:mainDocumentType w:val="formLetters" />
  …
</w:mailMerge>
```

In this example, the source document is of the `formLetters` type, as specified by the mainDocumentType element's val attribute being equal to `formLetters`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
catalog (Catalog Source Document)	Specifies that the mail merge source document is of the catalog type.
email (E-Mail Source Document)	Specifies that the mail merge source document is of the e-mail message type.
envelopes (Envelope Source Document)	Specifies that the mail merge source document is of the envelope type.
fax (Fax Source Document)	Specifies that the mail merge source document is of the fax type.
formLetters (Form Letter Source Document)	Specifies that the mail merge source document is of the form letter type.
mailingLabels (Mailing Label Source Document)	Specifies that the mail merge source document is of the mailing label type.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MailMergeDocType) is located in §A.1. *end note*]

17.18.55 ST_MailMergeOdsoFMDFieldType (Merge Field Mapping Types)

This simple types specifies the possible types used to indicate if a given mail merge field has been mapped to a column in the given external data source.

[*Example*: Consider the WordprocessingML below:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<w:odso>
  …
  <w:fieldMapData>
    <w:type w:val="dbColumn" />
    <w:name w:val="Country" />
    <w:mappedName w:val="Country or Region" />
    <w:column w:val="9" />
    …
  </w:fieldMapData>
</w:odso>
```

In this example, the `country` column within the given external data source must be mapped to the mail merge field `Country or Region`, as specified by the type element's val attribute being equal to `dbColumn`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dbColumn (Field Mapping to Data Source Column)	Specifies that the mail merge field has been mapped to a column in the given external data source.
null (Field Not Mapped)	Specifies that the mail merge field has not been mapped to a column in the given external data source.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MailMergeOdsoFMDFieldType) is located in §A.1. *end note*]

17.18.56 ST_MailMergeSourceType (Mail Merge ODSO Data Source Types)

This simple type specifies the type of external data source to be connected to via as part of the ODSO connection information for this mail merge. This setting is purely a suggestion of the data source type which is being used for this mail merge, and can be ignored in favor of an alternative mechanism if one is present.

[*Example*: Consider the following WordprocessingML fragment for a mail merge source or merged document:

```
<w:type w:val="database" />
```

The type element's val attribute is equal to `database`, specifying that the given merged WordprocessingML document has been connected to an external data source via the ODSO settings, and that the resulting data source was a database. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
addressBook (Address Book Data Source)	Specifies that a given merged WordprocesingML document has been connected to an address book of contacts.
database (Database Data Source)	Specifies that a given merged WordprocessingML document has been connected to a database.
document1 (Alternate Document Format Data Source)	Specifies that a given merged WordprocessingML document has been connected to another document format supported by the producing application. The format of this document is application-defined and outside the scope of ISO/IEC 29500.
document2 (Alternate Document Format Data Source Two)	Specifies that a given merged WordprocessingML document has been connected to another document format supported by the producing application. The format of this document is application-defined and outside the scope of ISO/IEC 29500.
email (E-Mail Program Data Source)	Specifies that a given merged WordprocessingML document has been connected to an e-mail application.
legacy (Legacy Document Format Data Source)	Specifies that a given merged WordprocessingML document has been connected to a legacy document format supported by the producing application. The format of this legacy document is application-defined and outside the scope of ISO/IEC 29500.
master (Aggregate Data Source)	Specifies that a given merged WordprocessingML document has been connected to a data source which aggregates other data sources.
native (Native Data Souce)	Specifies that a given merged WordprocessingML document has been connected to another document format native to the producing application. The format of this document is application-defined and outside the scope of ISO/IEC 29500.
text (Text File Data Source)	Specifies that a given merged WordprocessingML document has been connected to a text file.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MailMergeSourceType) is located in §A.1. *end note*]

17.18.57 ST_Merge (Merged Cell Type)

This element specifies the way in which a cell shall be included in a merged group of cells (horizontally or vertically) within the parent table.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a table with three rows and two columns with the last column completely vertically merged:

The second cell in the first row starts a vertical merge that is completed in the last cell, resulting in the following WordprocessingML:

```
<w:tbl>
  …
  <w:tr>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:vMerge w:val="restart"/>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
  <w:tr>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      …
    </w:tc>
    <w:tc>
      <w:tcPr>
        <w:vMerge w:val="continue"/>
      </w:tcPr>
      …
    </w:tc>
  </w:tr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <w:tr>
      <w:tc>
        …
      </w:tc>
      <w:tc>
        …
      </w:tc>
      <w:tc>
        <w:tcPr>
          <w:vMerge w:val="continue"/>
        </w:tcPr>
        …
      </w:tc>
    </w:tr>
  </w:tbl>
```

The val attribute of type ST_Merge on the vMerge element defines the cells that are vertically merged, and how each cell is merged together. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
continue (Continue Merged Region)	Specifies that the current cell continues a previously existing merged group of cells in the parent table. If the previous cell in the document (horizontally or vertically) does not either begin or continue a set of merged cells, then this value shall be ignored (i.e. a group of merged cells shall start with a merge whose ST_Merge value is `restart`).
restart (Start/Restart Merged Region)	Specifies that the current cell starts (or restarts) a group of merged cells in the parent table. After this value, all following cells which have a value of `continue` shall be merged into this merged cell group.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Merge) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.18.58 ST_MultiLevelType (Numbering Definition Type)

This simple type specifies the possible types of numbering which can be defined by a given abstract numbering type. This information shall only be used by a consumer to determine user interface behaviors for this numbering definition, and shall not be used to limit the behavior of the list (i.e. a list with multiple levels marked as `singleLevel` shall not be prevented from using levels 2 through 9).

[*Example*: Consider the WordprocessingML below:

```
<w:abstractNum w:abstractNumId="8">
  …
  <w:multiLevelType w:val="singleLevel" />
  …
</w:abstractNum>
```

This abstract numbering definition is specified to be of the `singleLevel` type by the multiLevelType element. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hybridMultilevel (Hybrid Multilevel Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a multiple levels, each of a potentially different kind (bullets vs. level text).
multilevel (Multilevel Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a multiple levels, each of the same kind (bullets vs. level text).
singleLevel (Single Level Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a single level only.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MultiLevelType) is located in §A.1. *end note*]

17.18.59 ST_NumberFormat (Numbering Format)

This simple type specifies the numbering format which shall be used for a group of automatically numbered objects,

[*Example*: A value of `lowerLetter` for page numbering indicates that a consumer must use lowercase letters for each page in this section: a,b,c… *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aiueo (AIUEO Order Half-Width Katakana)	Specifies that the sequence shall consist of one or more occurrences of a single half-width Katakana character from the set listed below, in the traditional a-i-u-e-o order. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–46 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–46 is U+FF71–U+FF9C, U+FF66, and U+FF9D, respectively. For values greater than 46, the sequence begins again, iterating through the same 46 values, repeating this pattern as necessary. [*Example*: The numbering for these items should be represented by the following pattern: ｱ, ｲ, ｳ, …, ﾜ, ﾝ, ｱｱ, ｲｲ, ｳｳ, … *end example*]
aiueoFullWidth (AIUEO Order Full-Width Katakana)	Specifies that the sequence shall consist of one or more occurrences of a single full-width katakana character from the set listed below, in the traditional a-i-u-e-o order. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–46 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–46 is U+30A2, U+30A4, U+30A6, U+30A8, U+30AA, U+30AB, U+30AD, U+30AF, U+30B1, U+30B3, U+30B5, U+30B7, U+30B9, U+30BB, U+30BD, U+30BF,U+30C1,U+30C4, U+30C6, U+30C8, U+30CA, U+30CB, U+30CC, U+30CD, U+30CE, U+30CF, U+30D2, U+30D5, U+30D8, U+30DB, U+30DE, U+30DF,U+30E0, U+30E1, U+30E2, U+30E4, U+30E6, U+30E8, U+30E9, U+30EA, U+30EB, U+30EC, U+30ED,U+30EF, U+30F0, U+30F1, U+30F2, and U+30F3, respectively.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	For values greater than 46, the sequence begins again, iterating through the same 46 values, repeating this pattern as necessary. [*Example*: The numbering for the items should be represented by the following pattern: ア, イ, ウ, …, ヲ, ン, アア, イイ, ウウ, … *end example*]
arabicAbjad (Arabic Abjad Numerals)	Specifies that the sequence shall consist of one or more occurrences of a single ascending Abjad numeral from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–28 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–28 is U+0623, U+0628, U+062C, U+062F, U+0647, U+0648, U+0632, U+062D, U+0637, U+064A, U+0643, U+0644, U+0645, U+0646, U+0633, U+0639, U+0641, U+0635, U+0642, U+0631, U+0634, U+062A, U+062B, U+062E,U+0630, U+0636,U+063A, and U+0638, respectively. For values greater than 28, the text displayed shall be constructed as follows: 1. Repeatedly subtract the size of the set (28) from the value until the result is equal to or less than the size of the set. 2. The remainder determines which character to use from the set above, and that character is written once and then written once and then written once and then repeated the number of times the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: أ, ب, ج, …, ظ, غ, أأ, ب ب, ج ج, …, ظ ظ, غ غ, أ أ أ, ب ب ب, ج ج ج, … *end example*]
arabicAlpha (Arabic Alphabet)	Specifies that the sequence shall consist of one or more occurrences of a single character in the Arabic alphabet from the set listed below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–28 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–28 is U+0623, U+0628, U+062A, U+062B, U+062C, U+062D, U+062E, U+062F, U+0630, U+0631, U+0632, U+0633, U+0634, U+0635, U+0636, U+0637, U+0638, U+0639, U+063A, U+0641, U+0642, U+0643, U+0644, U+0645, U+0646, U+0647, U+0648, and U+064A, respectively. For values greater than 28, the text displayed shall be constructed as follows: 1. Repeatedly subtract the size of the set (28) from the value until the result is equal to or less than the size of the set. 2. The remainder determines which character to use from the set above, and that character is written once and then written once and then written once and then repeated the number of times the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: أ, ب, ت, ..., و, ي, أ أ, ب ب, ت ت, ..., و و, ي ي, أأأ *end example*]
bahtText (Thai Baht Text)	Specifies that the sequence shall consist of a numeric value from the Thai counting system, with บาทถ้วน appended to the result. The portion of the sequence which is prepended to the static text shall be the equivalent value in the thaiCounting format, defined below. [*Example*: The values for the items should be represented by the following pattern: หนึ่งบาทถ้วน, สองบาทถ้วน, สามบาทถ้วน. *end example*]
bullet (Bullet)	Specifies that the sequence shall consist of the bullet character defined by the lvlText element (§17.9.11).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	[*Example*: ● *end example*]
cardinalText (Cardinal Text)	Specifies that the sequence shall consist of cardinal text of the run language. This sequence is a set of strings each of which is the textual representation, in the language of the lang element (§17.3.2.20), of a different unique position in that sequence. [*Example*: The numbering for the items in Spanish should be represented by the following pattern: Uno, Dos, Tres, …, Nueve, Diez, Once, … Diecineuve, Veinte, Veintiuno, … *end example*]
chicago (Chicago Manual of Style)	Specifies that the sequence shall consist of one or more occurrences of a single character from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–4 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–4 is U+002A, U+2020, U+2021, and U+00A7, respectively. For values greater than 4, the text displayed shall be constructed as follows: 1. Repeatedly subtract the size of the set (4) from the value until the result is equal to or less than the size of the set. 2. The remainder determines which character to use from the set above, and that character is written once and then written once and then repeated the number of times the size of the set was subtracted from the original value. [*Example*: The first nine items in this format are: *, †, ‡, §, **, ††, ‡‡, §§, ***. *end example*]
chineseCounting (Chinese Counting System)	Specifies that the sequence shall consist of one or more occurrences of a single ascending number from the Chinese counting system, from the set listed below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–10 and then repeats those same characters using the logic defined below to construct all other values. 〇 represents the number zero. The set of characters used by this numbering format for values 0–10 is U+25CB, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, U+4E5D, and U+5341, respectively. For values greater than 10, the text displayed shall be constructed as follows: 1. Divide the value by 10 and write the symbol that represents the remainder. If the quotient is less than 10, then write 十 to the left of the symbol, which represents the remainder. 2. Divide the quotient of the previous division by 10, and write the symbol, which represents the remainder, to the left of the existing characters. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 九, 十, 十一, 十二, …, 十九, 二十, 二十一, …, 九十九, 一〇〇, 一〇一, … *end example*]
chineseCountingThousand (Chinese Counting Thousand System)	Specifies that the sequence shall consist of one or more occurrences of a single sequential number from the Chinese counting thousand system, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–10, 100, 1,000, and 10,000, and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format is U+96F6, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, U+4E5D, U+5341, U+767E, U+5343, and U+4E07.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>
To construct a value that is beyond the set, but less than one hundred thousand, work from largest groups to smallest following these steps:

Create as many groups as possible that contain ten thousand in each group.

Write down the symbol representing that value (1–9):
<table>
<tr><th>Digit</th><th>Character</th></tr>
<tr><td>1</td><td>一 (U+4E00)</td></tr>
<tr><td>2</td><td>二 (U+4E8C)</td></tr>
<tr><td>3</td><td>三 (U+4E09)</td></tr>
<tr><td>4</td><td>四 (U+56DB)</td></tr>
<tr><td>5</td><td>五 (U+4E94)</td></tr>
<tr><td>6</td><td>六 (U+516D)</td></tr>
<tr><td>7</td><td>七 (U+4E03)</td></tr>
<tr><td>8</td><td>八 (U+516B)</td></tr>
<tr><td>9</td><td>九 (U+4E5D)</td></tr>
</table>

If no groups are formed, do not write any characters.
If groups were formed, write down the symbol representing ten thousand: 万

Repeat step 1 for groups of one thousand (千) using the corresponding symbol to indicate the groups (so five thousand would be 五千).

If the original value was between 10,000 and 100,000 and If no groups are formed (and the number is not a multiple of thousand) write the symbol 零 instead (so ten thousand and five would be 一万零五)

Repeat step 1 for groups of one hundred (百) using the corresponding symbol to indicate the groups (so five hundred would be 五百).

If the original value was between 1,000 and 10,000 and if no groups are formed (and the number is not a multiple of hundred) write the symbol 零 instead (so one thousand and five would be 一千零五)

</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	4. Repeat step 1 for groups of ten (十) using the corresponding symbol to indicate the groups (so fifty would be 五十). • If the original value was between 100 and 1000 and if no groups are formed (and the number is not a multiple of ten) write the symbol 零 instead (so one hundred and five would be 一百零五) 5. Write down the symbol for the remaining number. If the number is larger than one hundred thousand but less than one hundred million, perform the cycle for the numbers above one thousand, but use two characters to represent each group. So, for example, groups of one hundred thousand are represented as ten ten thousands (一十万). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands). [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, 二, 三, …, 八, 九, 一十, 一十一, 一十二, …, 一十九, 二十, 二十一, …, 九九, 一百, 一百一, … *end example*]
chineseLegalSimplified (Chinese Legal Simplified Format)	Specifies that the sequence shall consist of one or more occurrences of a single sequential number from the Chinese simplified legal format, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those are combined with additional characters to represent the corresponding power of ten. The set of characters used by this numbering format for values 1–10 is U+96F6, U+58F9, U+8D30, U+53C1, U+8086, U+4F0D, U+9646, U+67D2, U+634C, and U+7396, respectively. To construct a number that is less than one hundred thousand, work from largest groups to smallest following these steps: 1. Create as many groups as possible that contain

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>
ten thousand in each group.

Write down the symbol representing that value (1–9):
<table>
<tr><th>Digit</th><th>Character</th></tr>
<tr><td>1</td><td>壹(U+58F9)</td></tr>
<tr><td>2</td><td>贰 (U+8D30)</td></tr>
<tr><td>3</td><td>叁 (U+53C1)</td></tr>
<tr><td>4</td><td>肆(U+8086)</td></tr>
<tr><td>5</td><td>伍 (U+4F0D)</td></tr>
<tr><td>6</td><td>陆 (U+9646)</td></tr>
<tr><td>7</td><td>柒 (U+67D2)</td></tr>
<tr><td>8</td><td>捌 (U+634C)</td></tr>
<tr><td>9</td><td>玖 (U+7396)</td></tr>
</table>

If no groups are formed, do not write any characters.
If groups were formed, write down the symbol representing ten thousand: 万

2. Repeat step 1 for groups of one thousand (仟) using the corresponding symbol to indicate the groups (so five thousand would be 伍仟).

If the original value was between 10,000 and 100,000 and If no groups are formed (and the number is not a multiple of thousand) write the symbol 零 instead (so ten thousand and five would be 壹万零伍)

3. Repeat step 1 for groups of one hundred (佰) using the corresponding symbol to indicate the groups (so five hundred would be 伍佰).

If the original value was between 1,000 and 10,000 and if no groups are formed (and the number is not a multiple of hundred) write the symbol 零 instead (so one thousand and five would be 壹仟零伍)

4. Repeat step 1 for groups of ten (拾) using the corresponding symbol to indicate the groups (so fifty would be 伍拾).

If the original value was between 100 and 1000 and If no groups are formed (and the number is not a multiple of ten) write the symbol 零 instead (so one hundred and

</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	five would be 壹佰零伍) 5. Write down the symbol for the remaining number. If the number is larger than one hundred thousand but less than one hundred million, perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands (佰万). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands). [*Example*: The numbering for the items should be represented by the following pattern: 壹, 贰叁, ..., 捌, 玖, 壹拾, 壹拾壹, 壹拾贰, ..., 壹拾玖, 贰拾, 贰拾壹, ..., 玖玖, 壹佰, 壹佰零壹, … *end example*]
chosung (Korean Chosung Numbering)	Specifies that the sequence shall consist of one or more occurrences of a single sequential number from the Korean Chosung format, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–14 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–14 is U+3131, U+3134, U+3137, U+3139, U+3141, U+3142, U+3145, U+3147, U+3148, U+314A, U+314B, U+314C, U+314D, and U+314E, respectively. For values greater than 14, the text displayed shall be constructed as follows: 1. Repeatedly subtract the size of the set (14) from the value until the result is equal to or less than the size of the set. 2. The remainder determines which character to use from the set above, and that character is written once and then repeated the number of times the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	represented by the following pattern: ㄱ, ㄴ, ㄷ, …, ㅍ, ㅎ, ㄱㄱ, ㄴㄴ, ㄷㄷ, … *end example*]
custom (Custom Defined Number Format)	Specifies a custom number format using the parent element's attribute to specify a number format using the mechanism defined by the XSLT format attribute.
decimal (Decimal Numbers)	Specifies that the sequence shall consist of decimal numbering. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 0 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+0030–U+0039, respectively. Continue the sequence by using the following steps: 1. Increment the rightmost position. 2. Every time the end of the set is reached, for a given position, increment the position to the immediate left (if there is no position to the immediate left, create a new position and start the sequence of the new position at 1) and reset the current position to 0. [*Example*: The numbering for the items should be represented by the following pattern: 1, 2, 3, …, 8, 9, 10, 11, 12, …, 18, 19, 20, 21, … *end example*]
decimalEnclosedCircle (Decimal Numbers Enclosed in a Circle)	Specifies that the sequence shall consist of decimal numbering enclosed in a circle, using the enclosed character. This system uses a set of characters to represent the numbers 1–20. The set of characters used by this numbering format for values 1–20 is U+2460–U+2473, respectively. For values greater than the size of the set, the items fall back to the decimal format. [*Example*: The numbering for the items should be represented by the following pattern: ①, ②, ③, …, ⑲

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	, ⑳, 21, … *end example*]
decimalEnclosedCircleChinese (Decimal Numbers Enclosed in a Circle)	Identical to decimalEnclosedCircle.
decimalEnclosedFullstop (Decimal Numbers Followed by a Period)	Specifies that the sequence shall consist of decimal numbering followed by a period, using the appropriate character, as described below. This system uses a set of characters to represent the numbers 1–20. The set of characters used by this numbering format for values 1–20 is U+2488–U+249B, respectively. For values greater than the size of the set, the items fall back to the decimal format. [*Example*: The numbering for the items should be represented by the following pattern: 1., 2., 3., …, 19., 20., 21, … *end example*]
decimalEnclosedParen (Decimal Numbers Enclosed in Parenthesis)	Specifies that the sequence shall consist of decimal numbering enclosed in parentheses, using the appropriate character, as described below. This system uses a set of characters to represent the numbers 1–20. The set of characters used by this numbering format for values 1–20 is U+2474–U+2487, respectively. For values greater than the size of the set, the items fall back to the decimal format. [*Example*: The numbering for the items should be represented by the following pattern: ⑴, ⑵, ⑶, …, ⑲, ⑳, 21, 22, … *end example*]
decimalFullWidth (Full Width Arabic Numerals)	Specifies that the sequence shall consist of full-width Arabic numbering. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and ０ (represents the number zero) to construct the remaining values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	The set of characters used by this numbering format for values 0–9 is U+FF10–U+FF19, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: １, ２, ３, …, ８, ９, １０, １１, １２, …, １８, １９, ２０, ２１, … *end example*]
decimalHalfWidth (Half Width Arabic Numerals)	Specifies that the sequence shall consist of half-width Arabic numbering. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 0 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+0030–U+0039, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 1, 2, 3, …, 8, 9,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	10, 11, 12, …, 18, 19, 20, 21, … *end example*]
decimalZero (Initial Zero Arabic Numerals)	Specifies that the sequence shall consist of Arabic numbering with a zero added to numbers one through nine. To determine the text that is displayed for any value, this sequence specifies a set of paired characters (zero followed by an additional symbol) that represent positions 1–9 and then those same characters are combined with each other to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+0030–U+0039. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 01, 02, 03, …, 08, 09, 10, 11, 12, …, 18, 19, 20, 21, 22, …, 98, 99, 100, 101, … *end example*]
dollarText (Dollar Text)	Specifies that the sequence shall consist of a cardinal text value of the run language, with "`and 00/100`" (also in the run language) appended to the result. [*Note*: The latter text is fixed because values in numbering sequences are integer-based. *end note*] The cardinal text value which is prepended to the static text shall be the equivalent value in the cardinalText format, defined below. [*Example*: one and 00/100, two and 00/100, three and 00/100. *end example*]
ganada (Korean Ganada Numbering)	Specifies that the sequence shall consist of one or more occurrences of a single sequential number from

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the Korean Ganada format, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–14 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–14 is U+AC00, U+B098, U+B2E4, U+B77C, U+B9C8, U+BC14, U+C0AC, U+C544, U+C790, U+CC28, U+CE74, U+D0C0, U+D30C, and U+D558, respectively. For values greater than 14, the text displayed shall be constructed as follows: 1. Repeatedly subtract the size of the set (14) from the value until the result is equal to or less than the size of the set. 2. The remainder determines which character to use from the set above, and that sequence of character is repeated the number of times the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: 가, 나, 다, …, 파, 하, 가가, 나나, 다다, … *end example*]
hebrew1 (Hebrew Letters)	Specifies that the sequence shall consist of Hebrew letters from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9, each multiple of ten (less than 100), each multiple of 100 (less than 1000), etc. which are then combined with each other to construct the remaining values. The set of characters used by this numbering format is U+05D0–U+05D9, U+05DB, U+05DC, U+05DE, U+05E0–U+05E2, U+05E4, U+05E6–U+05EA, U+05E7–U+05E9, U+05EA, U+05DA, U+05DD, U+05DF, U+05E3, and U+05E5, respectively. To construct the text for any value, convert it from its decimal equivalent following these steps (writing right

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>
to left):

Replace the thousands digit with the appropriate symbol:
<table>
<tr><th>Digit</th><th>Symbol</th></tr>
<tr><td>1</td><td>א (U+05D0)</td></tr>
<tr><td>2</td><td>ב (U+05D1)</td></tr>
<tr><td>3</td><td>ג (U+05D2)</td></tr>
<tr><td>4</td><td>ד (U+05D3)</td></tr>
<tr><td>5</td><td>ה (U+05D4)</td></tr>
<tr><td>6</td><td>ו (U+05D5)</td></tr>
<tr><td>7</td><td>ז (U+05D6)</td></tr>
<tr><td>8</td><td>ח (U+05D7)</td></tr>
<tr><td>9</td><td>ט (U+05D8)</td></tr>
</table>

Replace the hundreds digit with the appropriate symbol:
<table>
<tr><th>Digit</th><th>Symbol</th></tr>
<tr><td>1</td><td>ק (U+05E7)</td></tr>
<tr><td>2</td><td>ר (U+05E8)</td></tr>
<tr><td>3</td><td>ש (U+ 05E9)</td></tr>
<tr><td>4</td><td>ת (U+05EA)</td></tr>
<tr><td>5</td><td>ך (U+05DA)</td></tr>
<tr><td>6</td><td>ם (U+05DD)</td></tr>
<tr><td>7</td><td>ן (U+05DF)</td></tr>
<tr><td>8</td><td>ף (U+05E3)</td></tr>
<tr><td>9</td><td>ץ (U+05E5)</td></tr>
</table>

If the remainder is 15 or 16, replace them as follows and stop:
<table>
<tr><th>Value</th><th>Symbols</th></tr>
<tr><td>15</td><td>טו (U+05D8, U+05D5)</td></tr>
<tr><td>16</td><td>טז (U+05D8, U+05D6)</td></tr>
</table>

Otherwise, replace the tens digit with the appropriate symbol:

</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Digit</th><th>Symbol</th></tr>
<tr><td>1</td><td>י (U+05d9)</td></tr>
<tr><td>2</td><td>כ (U+05DB)</td></tr>
<tr><td>3</td><td>ל (U+ 05DC)</td></tr>
<tr><td>4</td><td>מ (U+05DE)</td></tr>
<tr><td>5</td><td>נ (U+05E0)</td></tr>
<tr><td>6</td><td>ס (U+05E1)</td></tr>
<tr><td>7</td><td>ע (U+05E2)</td></tr>
<tr><td>8</td><td>פ (U+05E4)</td></tr>
<tr><td>9</td><td>צ (U+05E6)</td></tr>
</table>
5. Replace the ones digit with the appropriate symbol:
<table>
<tr><th>Digit</th><th>Symbol</th></tr>
<tr><td>1</td><td>א (U+05d0)</td></tr>
<tr><td>2</td><td>ב (U+05d1)</td></tr>
<tr><td>3</td><td>ג (U+ 05d2)</td></tr>
<tr><td>4</td><td>ד (U+05d3)</td></tr>
<tr><td>5</td><td>ה (U+05d4)</td></tr>
<tr><td>6</td><td>ו (U+05D5)</td></tr>
<tr><td>7</td><td>ז (U+05D6)</td></tr>
<tr><td>8</td><td>ח (U+05d7)</td></tr>
<tr><td>9</td><td>ט (U+05D8)</td></tr>
</table>
6. Reordering a number group will not change its value. If a number spells out a Hebrew word with a negative or positive connotation, the number group can be rearranged.

[*Example*: The numbering for the items should be represented by the following pattern: א, ב, ג, … טז, זי, חי, טי, כ, אכ, … *end example*]</td></tr>
<tr><td>hebrew2 (Hebrew Alphabet)</td><td>Specifies that the sequence shall consist of the Hebrew alphabet.

To determine the text that is displayed for any value, this sequence specifies a set of characters that</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	represent positions 1–22 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–22 is U+05D0–U+05D9, U+05DB, U+05DC, U+05DE, U+05E0–U+05E2, U+05E4, and U+05E6–U+05EA, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (22) from the value until the result is equal to or less than the size of the set. 2. Write the symbol represented by the result value. 3. Then the ת symbol is repeated (to the right of the first symbol) for each time the size of the set was subtracted from the original value. 4. Reordering a number group does not change its value. If a number spells out a Hebrew word with a negative or positive connotation, the number group can be rearranged. [*Example*: The numbering for the items should be represented by the following pattern: א, ב, ג, …, את, בת, … *end example*]
hex (Hexadecimal Numbering)	Specifies that the sequence shall consist of hexadecimal numbering. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–15 and then those same characters are combined with each other and 0 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–15 is U+0030–U+0039, and U+0041–U+0046, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 16 and write the symbol which represents the remainder.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	2. Divide the quotient of the previous division by 16 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 1, 2, 3, …, E, F, 10,11, 12, …, 1E, 1F, 20, 21, … *end example*]
hindiConsonants (Hindi Consonants)	Specifies that the sequence shall consist of one or more occurrences of a single Hindi consonant from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–18 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–18 is U+0905–U+0914, U+0905 combined with U+0902, and U+0905 combined with U+0903, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (18) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use, and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: अ, आ, इ, …, अं,अः, अअ, आआ, इइ, …, अंअं, अःअः, अअअ, आआआ, इइइ, … *end example*]
hindiCounting (Hindi Counting System)	Specifies that the sequence shall consist of sequential numbers from the Hindi counting system. This sequence is a set of strings each of which is the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	full name, in Hindi, of the next value in that sequence. [*Example*: The numbering for the items should be represented by the following pattern: एक, दो, तीन, चार, पाँच, छः, सात, आठ, नौ, दस, … *end example*]
hindiNumbers (Hindi Numbers)	Specifies that the sequence shall consist of one or more occurrences of a single Hindi number from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and ० (U+0966, which represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 1–9 is U+0967, U+0968, U+0969, U+096A, U+096B, U+096C, U+096D, U+096E, and U+096F, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: १, २, ३, …, ८, ९, १०, ११, १२, …, १८, १९, २०, २१, … *end example*]
hindiVowels (Hindi Vowels)	Specifies that the sequence shall consist of one or more occurrences of a single Hindi vowel from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–37 and then repeats those same

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–37 is U+0915–U+0939, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (37) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use, and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: क, ख, ग, …, स, ह, ककक, खख, गग, …, सस, हह, ककक, खखख, गगग, … *end example*]
ideographDigital (Ideographs)	Specifies that the sequence shall consist of sequential numerical ideographs, using the appropriate character, as described below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 〇 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+3007, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, and U+4E5D, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 八, 九, 一〇, 一一, 一二, …, 一八, 一九, 二〇, 二一, … *end example*]
ideographEnclosedCircle (Ideographs Enclosed in a Circle)	Specifies that the sequence shall consist of sequential numerical ideographs, using the appropriate character, as described below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–10. The set of characters used by this numbering format for values 1–10 is U+3220–U+3229, respectively. For values greater than the size of the set, the items fall back to the decimal format. [*Example*: The numbering for the items should be represented by the following pattern: ㈠, ㈡, ㈢, …, ㈨, ㈩, 11,12, … *end example*]
ideographLegalTraditional (Traditional Legal Ideograph Format)	Specifies that the sequence shall consist of sequential numerical traditional legal ideographs. This system uses a set of characters to represent the numbers 1–9 and then those are combined with additional characters to represent the corresponding power of ten. The set of characters used by this numbering format is U+58F9, U+8CB3, U+53C3, U+8086, U+4F0D, U+9678, U+67D2, U+634C, U+7396, U+62FE, U+4F70, U+4EDF,and U+842C, respectively. To construct a number that is less than hundred thousand, you work from largest groups to smallest following these steps: 1. Create as many groups as possible that contain ten thousand in each group. • Write down the symbol representing that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>value (1–9):
<table>
<tr><th>Digit</th><th>Character</th></tr>
<tr><td>1</td><td>壹 (U+58F9)</td></tr>
<tr><td>2</td><td>貳 (U+8CB3)</td></tr>
<tr><td>3</td><td>參 (U+53C3)</td></tr>
<tr><td>4</td><td>肆 (U+8086)</td></tr>
<tr><td>5</td><td>伍 (U+4F0D)</td></tr>
<tr><td>6</td><td>陸 (U+9678)</td></tr>
<tr><td>7</td><td>柒 (U+67D2)</td></tr>
<tr><td>8</td><td>捌 (U+634C)</td></tr>
<tr><td>9</td><td>玖 (U+7396)</td></tr>
</table>

If no groups are formed, do not write any characters.
If groups were formed, write down the symbol representing ten thousand (the power of ten represented by that position): 萬

2. Repeat this for groups of one thousand (仟), one hundred (佰) and ten (拾) using the corresponding symbol to indicate the groups (so five hundred would be 伍佰 and fifty would be 伍拾).
3. Write down the symbol for the remaining number.

If the number is larger than hundred thousand but less than one hundred million, you perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands (拾萬). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands).

[Example: The numbering for the items should be represented by the following pattern: 壹, 貳, 參, …, 捌, 玖, 壹拾, 壹拾壹, 壹拾貳, …, 壹拾玖, 貳拾, 貳拾壹, … end example]</td></tr>
<tr><td>ideographTraditional (Traditional Ideograph Format)</td><td>Specifies that the sequence shall consist of sequential numerical traditional ideographs.</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	This system uses a set of characters (elements in the Chinese Sexagenary cycle) to represent the numbers 1–10. For values greater than the size of the set, the items fall back to the decimal format. The set of characters used by this numbering format for values 1–10 is U+7532, U+4E59, U+4E19, U+4E01, U+620A, U+5DF1, U+5E9A, U+8F9B, U+58EC, and U+7678, respectively. [*Example*: The numbering for the items should be represented by the following pattern: 甲, 乙, 丙, 丁, …, 壬, 癸, 11, 12, … *end example*]
ideographZodiac (Zodiac Ideograph Format)	Specifies that the sequence shall consist of sequential zodiac ideographs. This system uses a set of characters (animals in the Chinese Sexagenary cycle) to represent the numbers 1–12. For values greater than the size of the set, the items fall back to the decimal format. The set of characters used by this numbering format for values 1–12 is U+5B50, U+4E11, U+5BC5, U+536F, U+8FB0, U+5DF3, U+5348, U+672A, U+7533, U+9149, U+620C, and U+4EA5, respectively. [*Example*: The numbering for the items should be represented by the following pattern: 子, 丑, 寅, …, 戌, 亥, 13, 14, … *end example*]
ideographZodiacTraditional (Traditional Zodiac Ideograph Format)	Specifies that the sequence shall consist of sequential traditional zodiac ideographs. This system uses a set of character pairs (all of the element-animal combinations of the Chinese sexagenary cycle) to represent the numbers 1–60 and then those same character pairs are repeated to construct the remaining values. The set of character pairs used by this numbering format for values 1–60 is U+7532, U+5B50; U+4E59, U+4E11; U+4E19, U+5BC5; U+4E01, U+536F; U+620A, U+8FB0; U+5DF1, U+5DF3; U+5E9A, U+5348; U+8F9B, U+672A; U+58EC, U+7533; U+7678, U+9149; U+7532, U+620D; U+4E59, U+4EA5; U+4E19, U+5B50; U+4E01, U+4E11; U+620A, U+5BC5; U+5DF1, U+536F; U+5E9A,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	U+8FB0; U+8F9B, U+5DF3; U+58EC, U+5348; U+7678, U+672A; U+7532, U+7533; U+4E59, U+9149; U+4E19, U+620D; U+4E01, U+4EA5; U+620A, U+5B50; U+5DF1, U+4E11; U+5E9A, U+5BC5; U+8F9B, U+536F; U+58EC, U+8FB0; U+7678, U+5DF3; U+7532, U+5348; U+4E59, U+672A; U+4E19, U+7533; U+4E01, U+9149; U+620A, U+620D; U+5DF1, U+4EA5; U+5E9A, U+5B50; U+8F9B, U+4E11; U+58EC, U+5BC5; U+7678, U+536F; U+7532, U+8FB0; U+4E59, U+5DF3; U+4E19, U+5348; U+4E01, U+672A; U+620A, U+7533; U+5DF1, U+9149; U+5E9A, U+620D; U+8F9B, U+4EA5; U+58EC, U+5B50; U+7678, U+4E11; U+7532, U+5BC5; U+4E59, U+536F; U+4E19, U+8FB0; U+4E01, U+5DF3; U+620A, U+5348; U+5DF1, U+672A; U+5E9A, U+7533; U+8F9B, U+9149; U+58EC, U+620D; U+7678, U+4EA5, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (60) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character pair to use. [*Example*: The numbering for the items should be represented by the following pattern: 甲子, 乙丑, 丙寅, …, 壬戌, 癸亥, 甲子, 乙丑, 丙寅, … *end example*]
iroha (Iroha Ordered Katakana)	Specifies that the sequence shall consist of the iroha. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–48 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–48 is U+FF72, U+FF9B, U+FF8A, U+FF86, U+FF8E, U+FF8D, U+FF84, U+FF81, U+FF98, U+FF87, U+FF99,U+FF66, U+FF9C, U+FF76, U+FF96, U+FF80, U+FF9A, U+FF7F, U+FF82, U+FF88, U+FF85, U+FF97, U+FF91, U+FF73, U+30F0, U+FF89, U+FF75, U+FF78, U+FF94, U+FF8F, U+FF79, U+FF8C, U+FF7A, U+FF74, U+FF83, U+FF71, U+FF7B, U+FF77, U+FF95, U+FF92, U+FF90, U+FF7C, U+30F1, U+FF8B, U+FF93, U+FF7E, U+FF7D, and U+FF9D, respectively.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (48) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use. [*Example*: The numbering for the items should be represented by the following pattern: ｲ, ﾛ, ﾊ, …, ｽ, ﾝ, ｲ, ﾛ, ﾊ, … *end example*]
irohaFullWidth (Full-Width Iroha Ordered Katakana)	Specifies that the sequence shall consist of the full-width forms of the iroha. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–48 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–48 is U+30A4, U+30ED, U+30CF, U+30CB, U+30DB, U+30D8, U+30C8, U+30C1, U+30EA, U+30CC, U+30EB, U+30F2, U+30EF, U+30AB, U+30E8, U+30BF, U+30EC, U+30BD, U+30C4, U+30CD, U+30CA, U+30E9, U+30E0, U+30A6, U+30F0, U+30CE, U+30AA, U+30AF, U+30E4, U+30DE, U+30B1, U+30D5, U+30B3, U+30A8, U+30C6, U+30A2, U+30B5, U+30AD, U+30E6, U+30E1, U+30DF, U+30B7, U+30F1, U+30D2, U+30E2, U+30BB, U+30B9, and U+30F3, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (48) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use. [*Example*: The numbering for the items should be represented by the following pattern: イ, ロ, ハ, …, ス, ン, イ, ロ, ハ, … *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td>japaneseCounting (Japanese Counting System)</td><td>
Specifies that the sequence shall consist of sequential numbers from the Japanese counting system.

This system uses a set of characters to represent the numbers 1–9 and then those are combined with additional characters to represent the corresponding power of ten.

The set of characters used by this numbering format is U+3007, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, U+4E5D, U+5341, U+5343, and U+767E, respectively.

To construct a number that is less than ten thousand, you work from largest groups to smallest following these steps:

1. Create as many groups as possible that contain one thousand in each group.
• Write down the symbol representing that value (1–9):
<table>
<tr><th>Digit</th><th>Character</th></tr>
<tr><td>1</td><td>一 (U+4E00)</td></tr>
<tr><td>2</td><td>二 (U+4E8C)</td></tr>
<tr><td>3</td><td>三 (U+4E09)</td></tr>
<tr><td>4</td><td>四 (U+56DB)</td></tr>
<tr><td>5</td><td>五 (U+4E94)</td></tr>
<tr><td>6</td><td>六 (U+516D)</td></tr>
<tr><td>7</td><td>七 (U+4E03)</td></tr>
<tr><td>8</td><td>八 (U+516B)</td></tr>
<tr><td>9</td><td>九 (U+4E5D)</td></tr>
</table>
• If no groups are formed, do not write any characters.
• If groups were formed, write down the symbol representing one thousand (the power of ten represented by that position): 千
2. Repeat this for groups of one hundred (百) and ten (十) using the corresponding symbol to indicate the groups (so five hundred would be 五百 and fifty would be 五十).
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	3. Write down the symbol for the remaining number. If the number is larger than ten thousand but less than one hundred million, you perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands (百万). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands). [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 八, 九, 十, 十一, 十二, …, 十八, 十九, 二十, 二十一, … *end example*]
japaneseDigitalTenThousand (Japanese Digital Ten Thousand Counting System)	Specifies that the sequence shall consist of sequential numbers from the Japanese digital ten thousand counting system. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 〇 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+3007, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, and U+4E5D, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>represented by the following pattern: 一, 二, 三, …, 八, 九, 一〇, 一一, 一二, …, 一八, 一九, 二〇, 二一, … end example]</td></tr>
<tr><td>japaneseLegal (Japanese Legal Numbering)</td><td>Specifies that the sequence shall consist of sequential numbers from the Japanese legal counting system.

This system uses a set of characters to represent the numbers 1–9 and then those are combined with additional characters to represent the corresponding power of ten.

The set of characters used by this numbering format for values is U+58F1, U+5F10, U+53C2, U+56DB, U+4F0D, U+516D, U+4E03, U+516B, U+4E5D, U+62FE, U+767E, U+842C, and U+9621, respectively.

To construct a number that is less than hundred thousand, you work from largest groups to smallest following these steps:

1. Create as many groups as possible that contain ten thousand in each group.
• Write down the symbol representing that value (1–9):
<table><tr><th>Digit</th><th>Character</th></tr><tr><td>1</td><td>壱(U+58F1)</td></tr><tr><td>2</td><td>弐(U+5F10)</td></tr><tr><td>3</td><td>参(U+53C2)</td></tr><tr><td>4</td><td>四(U+56DB)</td></tr><tr><td>5</td><td>伍(U+4F0D)</td></tr><tr><td>6</td><td>六(U+516D)</td></tr><tr><td>7</td><td>七(U+4E03)</td></tr><tr><td>8</td><td>八(U+516B)</td></tr><tr><td>9</td><td>九(U+4E5D)</td></tr></table>• If no groups are formed, do not write any characters.
• If groups were formed, write down the symbol representing ten thousand (the power of ten represented by that position): 萬
2. Repeat this for groups of one thousand (阡),</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>one hundred (百) and ten (拾) using the corresponding symbol to indicate the groups (so five hundred would be 伍百 and fifty would be 伍拾).
3. Write down the symbol for the remaining number.

If the number is larger than one hundred thousand but less than one hundred million, you perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands (拾萬). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands).

[Example: The numbering for the items should be represented by the following pattern: 壱, 弐, 参, …, 八, 九, 壱拾, 壱拾壱, 壱拾弐, … , 壱拾八, 壱拾九, 弐拾, 弐拾壱, … end example]</td></tr>
<tr><td>koreanCounting (Korean Counting System)</td><td>Specifies that the sequence shall consist of sequential numbers from the Korean counting system.

This system uses a set of characters to represent the numbers 1–9 and then those are combined with additional characters to represent the corresponding power of ten.

The set of characters used by this numbering format is U+C77C, U+C774, U+C0BC, U+C0AC, U+C624, U+C721, U+CE60, U+D314, U+AD6C, U+C2ED, U+B9CC, U+CC9C, and U+BC31, respectively.

To construct a number that is less than hundred thousand, you work from largest groups to smallest following these steps:

1. Create as many groups as possible that contain ten thousand in each group.
• Write down the symbol representing that value (1–9):
<table><tr><th>Digit</th><th>Character</th></tr><tr><td>1</td><td>일 (U+C77C)</td></tr><tr><td>2</td><td>이 (U+C774)</td></tr></table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>3</td><td>삼 (U+C0BC)</td></tr>
<tr><td>4</td><td>사 (U+C0AC)</td></tr>
<tr><td>5</td><td>오 (U+C624)</td></tr>
<tr><td>6</td><td>육 (U+C721)</td></tr>
<tr><td>7</td><td>칠 (U+CE60)</td></tr>
<tr><td>8</td><td>팔 (U+D314)</td></tr>
<tr><td>9</td><td>구 (U+AD6C)</td></tr>
</table>

If no groups are formed, do not write any characters.
If groups were formed, write down the symbol representing ten thousand (the power of ten represented by that position): 만

2. Repeat this for groups of one thousand (천), one hundred (백) and ten (십) using the corresponding symbol to indicate the groups (so five hundred would be 오백 and fifty would be 오십).
3. Write down the symbol for the remaining number.

If the number is larger than one hundred thousand but less than one hundred million, you perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands 십일만. An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands).

[Example: The numbering for the items should be represented by the following pattern: 일, 이, 삼, …, 팔, 구, 십, 십일, … end example]</td></tr>
<tr><td>koreanDigital (Korean Digital Counting System)</td><td>Specifies that the sequence shall consist of sequential numbers from the Korean digital counting system.

To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 영(represents the number zero) to construct the</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	remaining values. The set of characters used by this numbering format for values 0–9 is U+C601, U+C77C, U+C774, U+C0BC, U+C0AC, U+C624, U+C721, U+CE60, U+D314, and U+AD6C, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 일, 이, 삼, …, 팔, 구, 일영, 일일, … *end example*]
koreanDigital2 (Korean Digital Counting System Alternate)	Specifies that the sequence shall consist of sequential numbers from the Korean digital counting system. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 零 (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+96F6, U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, and U+4E5D, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>position.
3. Repeat step 2 until the remaining value is equal to zero.

[Example: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 八, 九, 一零, 一一, … end example]</td></tr>
<tr><td>koreanLegal (Korean Legal Numbering)</td><td>Specifies that the sequence shall consist of sequential numbers from the Korean legal numbering system.

This system uses a set of characters to represent the numbers 1–9 and then those are combined with additional characters which represent the multiples of ten (less than one hundred).

The set of characters used by this numbering format is U+D558, U+B098, U+B458, U+C14B, U+B137, U+B2E4, U+C12F, U+C5EC, U+C12F, U+C77C, U+ACF1, U+C5EC, U+B35F, U+C544, U+D649, U+C5F4, U+C2A4, U+BB3C, U+C11C, U+B978, U+B9C8, U+D754, U+C270, U+C608, U+C21C, U+C77C, U+D754, U+C5EC, U+B4E0, U+C544, and U+D754, respectively.

They are assigned according to the following table:

<table>
<tr><th>Digit</th><th>Characters</th></tr>
<tr><td>1</td><td>하나 (U+D558, U+B098)</td></tr>
<tr><td>2</td><td>둘 (U+B458)</td></tr>
<tr><td>3</td><td>셋 (U+C14B)</td></tr>
<tr><td>4</td><td>넷 (U+B137)</td></tr>
<tr><td>5</td><td>다섯 (U+B2E4, U+C12F)</td></tr>
<tr><td>6</td><td>여섯 (U+C5EC, U+C12F)</td></tr>
<tr><td>7</td><td>일곱 (U+C77C, U+ACF1)</td></tr>
<tr><td>8</td><td>여덟 (U+C5EC, U+B35F)</td></tr>
<tr><td>9</td><td>아홉 (U+C544 , U+D649)</td></tr>
<tr><td>10</td><td>열 (U+C5F4)</td></tr>
<tr><td>20</td><td>스물 (U+C2A4, U+BB3C)</td></tr>
<tr><td>30</td><td>서른 (U+C11C, U+B978)</td></tr>
<tr><td>40</td><td>마흔 (U+B9C8, U+D754)</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	<table><tr><td>50</td><td>쉰 (U+C270)</td></tr><tr><td>60</td><td>예순 (U+C608, U+C21C)</td></tr><tr><td>70</td><td>일흔 (U+C77C, U+D754)</td></tr><tr><td>80</td><td>여든 (U+C5EC, U+B4E0)</td></tr><tr><td>90</td><td>아흔 (U+C544, U+D754)</td></tr></table> [*Example*: The numbering for the items should be represented by the following pattern: 하나, 둘, 셋, …, 아홉, 열, 열하나, 열둘, …, 스물, 스물하나, … *end example*]
lowerLetter (Lowercase Latin Alphabet)	Specifies that the sequence shall consist of one or more occurrences of a single letter of the Latin alphabet in lower case from the set described below. This system uses a set of characters to represent the numbers 1 to the length of the language of the alphabet and then those same characters are combined to construct the remaining values. The characters used by for this numbering format is determined by using the language of the lang element (§17.3.2.20). Specifically: • When the script in use is derived from the Latin alphabet (a–z), that alphabet is used. [*Example*: For Norwegian (Nyorsk), the following Unicode characters are used by this numbering format: U+0061–U+007A, U+00E6, U+00F8, U+00E5. *end example*] • When the language in use is based on any other system, the characters U+0061–U+007A are used. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use and the same character is written once and then repeated for each time the size of the set was subtracted from the original value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: For English, the numbering for the items should be represented by the following pattern: a, b, c, …, y, z, aa, bb, cc, …, yy, zz, aaa, bbb, ccc, … *end example*]
lowerRoman (Lowercase Roman Numerals)	Specifies that the sequence shall consist of lowercase roman numerals. This system uses a set of characters to represent the numbers 1, 5, 10, 50, 100, 500, and 1000 and then those are combined with each other to construct the remaining values. The set of characters used by this numbering format is U+0069, U+0076, U+0078, U+006C, U+0063, U+0064, U+006D, respectively. To construct a number that is outside the set, you work from largest groups to smallest following these steps: 1. Create as many groups as possible that contain one thousand in each group. • The symbol representing one thousand (the power of ten represented by that position): m is repeated for the number of groups formed. If no groups are formed, do not write any symbol. 2. Repeat this for groups of nine hundred (cm), five-hundred (d), four-hundred (cd), one-hundred (c), ninety (xc), fifty (l), forty (xl), ten (x), nine (ix), five (v), four (iv) and finally one (i) using the corresponding symbol to indicate the groups (so four-hundred fifty would be cdl and forty-five would be xlv). [*Example*: The numbering for the items should be represented by the following pattern: i, ii, iii, iv, …, xviii, xix, xx, xxi, … *end example*]
none (No Numbering)	Specifies that the sequence shall not display any numbering.
numberInDash (Number With Dashes)	Specifies that the sequence shall consist of the Arabic numbering surrounded by hyphen-minus characters (U+002D). To determine the text that is displayed for any value,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and 0 (U+0030, which represents the number zero) to construct the remaining values. The set of characters used by this numbering format is U+002D (-) and, for values 1–9, U+0031–U+0039, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. 4. Place the final number between two dashes. [*Example*: The numbering for the items should be represented by the following pattern: - 1 -,- 2 -, - 3 -, …, - 8 -,- 9 -,- 10 -, - 11 -, - 12 -, …, - 18 -, - 19 -, - 20 -, - 21 -, … *end example*]
ordinal (Ordinal)	Specifies that the sequence shall consist of ordinals of the run language. This sequence is a set of strings each of which is the textual representation, in the language of the lang element (§17.3.2.20), of a different unique position in that sequence. [*Example*: The numbering for the items in French should be represented by the following pattern: 1er, 2e, 3e, …, 9e, 10e, 11e, … 19e, 20e, 21e, … *end example*]
ordinalText (Ordinal Text)	Specifies that the sequence shall consist of ordinal text of the run language. This sequence is a set of strings each of which is the textual representation, in the language of the lang element (§17.3.2.20), of a different unique position in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	that sequence. [*Example*: The numbering for the items in German should be represented by the following pattern: Erste, Zweite, Dritte, …, Neunte, Zehnte, Elfte, …, Neunzehnte, Zwanzigste, Einundzwanzigste, … *end example*]
russianLower (Lowercase Russian Alphabet)	Specifies that the sequence shall consist of one or more occurrences of a single letter of the Russian alphabet in lower case, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–29 and then repeats those same characters using the logic defined below to construct all other values. The set of characters used by this numbering format for values 1–29 is U+0430–U+0438, U+043A–U+043F, U+0440–U+0449, U+044B, U+044D, U+044E, and U+044F, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (29) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: а, б, в, …, ю, я, аа, бб, вв, …, юю, яя, ааа, ббб, ввв, … *end example*]
russianUpper (Uppercase Russian Alphabet)	Specifies that the sequence shall consist of one or more occurrences of a single letter of the Russian alphabet in upper case, from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–29 and then repeats those same characters using the logic defined below to construct all other values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	The set of characters used by this numbering format for values 1–29 is U+0410–U+0418, U+041A–U+041F, U+0420–U+0429, U+042B, U+042D, U+042E, and U+042F, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (29) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: А, Б, В, …, Ю, Я, АА, ББ, ВВ, …, ЮЮ, ЯЯ, ААА, БББ, ВВВ, … *end example*]
taiwaneseCounting (Taiwanese Counting System)	Specifies that the sequence shall consist of sequential numbers from the Taiwanese counting system. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and ○ (U+25CB, which represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 1–10 is U+4E00, U+4E8C,U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, U+4E5D, and U+5341, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. • If the quotient is less than 10 then write 十 to the left of the symbol, which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Enumeration Value</th><th>Description</th></tr>
<tr><td></td><td>the remainder, to the left of the existing positions.
3. Repeat step 2 until the remaining value is equal to zero.

[Example: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 九, 十, 十一, 十二, …, 十九, 二十, 二十一, …, 九十九, 一○○, 一○一, … end example]</td></tr>
<tr><td>taiwaneseCountingThousand (Taiwanese Counting Thousand System)</td><td>Specifies that the sequence shall consist of sequential numbers from the Taiwanese counting thousand system.

This system uses a set of characters to represent the numbers 1–10 and then those are combined with additional characters to construct the remaining characters.

The set of characters used by this numbering format is U+4E00, U+4E8C, U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, U+4E5D, U+842C, U+5343, U+5341, U+767E, and U+96F6.

To construct a number that is beyond the set but less than hundred thousand, you work from largest groups to smallest following these steps:

1. Create as many groups as possible that contain ten thousand in each group.
• Write down the symbol representing that value (1–9):
<table>
<tr><th>Digit</th><th>Character</th></tr>
<tr><td>1</td><td>一 (U+4E00)</td></tr>
<tr><td>2</td><td>二 (U+4E8C)</td></tr>
<tr><td>3</td><td>三 (U+4E09)</td></tr>
<tr><td>4</td><td>四 (U+56DB)</td></tr>
<tr><td>5</td><td>五 (U+4E94)</td></tr>
<tr><td>6</td><td>六 (U+516D)</td></tr>
<tr><td>7</td><td>七 (U+4E03)</td></tr>
<tr><td>8</td><td>八 (U+516B)</td></tr>
<tr><td>9</td><td>九 (U+4E5D)</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	• If no groups are formed, do not write any characters. • If groups were formed, write down the symbol representing ten thousand: 萬 2. Repeat step 1 this for groups of one thousand (千) using the corresponding symbol to indicate the groups (so five thousand would be 五千). • If the original value was between 10,000 and 100,000 and If no groups are formed (and the number is not a multiple of ten) write the symbol 零 instead (so ten thousand and five would be 一萬零五) 3. Repeat step 1 this for groups of one hundred (百) using the corresponding symbol to indicate the groups (so five hundred would be 五百). 4. The value 10 uses the symbol 十. For all other groups of ten, use the corresponding symbol to indicate the groups (so fifty would be 五十). • If the original value was between 100 and 1000 and If no groups are formed (and the number is not a multiple of ten) write the symbol 零 instead (so one hundred and five would be 一百零五) 5. Write down the symbol for the remaining number. If the number is larger than one hundred thousand but less than one hundred million, you perform the cycle for the numbers above one thousand but use two characters to represent each group. So, for example, groups of one million are represented as one hundred ten thousands (十萬). An additional symbol for counting groups is introduced at one hundred million (because 10 million is one thousand ten thousands). [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, 三, …, 八, 九, 一十, 十一, 十二, …, 十九, 二十, 二十一, …, 九九, 一百, 一百零一, … *end example*]
taiwaneseDigital (Taiwanese Digital Counting System)	Specifies that the sequence shall consist of sequential numbers from the Taiwanese digital counting system. To determine the text that is displayed for any value, this sequence specifies a set of characters that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	represent positions 1–9 and then those same characters are combined with each other and ○ (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+25CB, U+4E00, U+4E8C,U+4E09, U+56DB, U+4E94, U+516D, U+4E03, U+516B, and U+4E5D, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero. [*Example*: The numbering for the items should be represented by the following pattern: 一, 二, …, 八, 九, 一○,一一, 一二, …, 一八, 一九, 二○, 二一, … *end example*]
thaiCounting (Thai Counting System)	Specifies that the sequence shall consist of sequential numbers from the Thai counting system. This sequence is a set of strings each of which is the full name, in Thai, of the next value in that sequence. [*Example*: The numbering for the items should be represented by the following pattern: หนึ่ง, สอง, สาม, สี่, ห้า, หก, เจ็ด, แปด, เก้า, สิบ, … *end example*]
thaiLetters (Thai Letters)	Specifies that the sequence shall consist of one or more occurrences of a single Thai letter from the set listed below. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–41 and then repeats those same characters using the logic defined below to construct all other values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	The set of characters used by this numbering format for values 1–41 is U+0E01, U+0E02, U+0E04, U+0E07–U+0E23, U+0E25, and U+0E27–U+0E2E, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set (41) from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: The numbering for the items should be represented by the following pattern: ก, ข, ค, …, อ, ฮ, กก, ขข, คค, …, ออ, ฮฮ, กกก, ขขข, คคค, … *end example*]
thaiNumbers (Thai Numerals)	Specifies that the sequence shall consist of Thai numerals. To determine the text that is displayed for any value, this sequence specifies a set of characters that represent positions 1–9 and then those same characters are combined with each other and ๐ (represents the number zero) to construct the remaining values. The set of characters used by this numbering format for values 0–9 is U+0E50, U+0E51, U+0E52, U+0E53, U+0E54, U+0E55, U+0E56, U+0E57, U+0E58, and U+0E59, respectively. For values greater than the size of the set, the number is constructed by following these steps: 1. Divide the value by 10 and write the symbol which represents the remainder. 2. Divide the quotient of the previous division by 10 and write the symbol, which represents the remainder, to the left of the existing position. 3. Repeat step 2 until the remaining value is equal to zero.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: The numbering for the items should be represented by the following pattern: ๑, ๒, ๓, …, ๘, ๙, ๑๐, ๑๑, ๑๒, …, ๑๘, ๑๙, ๒๐, ๒๑, ๒๒, …, ๒๘, ๒๙, … *end example*]
upperLetter (Uppercase Latin Alphabet)	Specifies that the sequence shall consist of one or more occurrences of a single letter of the Latin alphabet in upper case, from the set listed below. This system uses a set of characters to represent the numbers 1 to the length of the language of the alphabet and then those same characters are combined to construct the remaining values. The characters used by this numbering format is determined by using the language of the lang element (§17.3.2.20). Specifically: • When the script in use is derived from the Latin alphabet (A–Z), that alphabet is used. [*Example*: For Norwegian (Nyorsk), the following Unicode characters are used by this numbering format: U+0041–U+005A, U+00C6, U+00D8, U+00C5. *end example*] • When the language in use is based on any other system, the characters U+0041–U+005A are used. For values greater than the size of the set, the number is constructed by following these steps: 1. Repeatedly subtract the size of the set from the value until the result is equal to or less than the size of the set. 2. The result value determines which character to use, and the same character is written once and then repeated for each time the size of the set was subtracted from the original value. [*Example*: For English, the numbering for the items should be represented by the following pattern: A, B, C, …, Y, Z, AA, BB, CC, …, YY, ZZ, AAA, BBB, CCC, … *end example*]
upperRoman (Uppercase Roman Numerals)	Specifies that the sequence shall consist of uppercase roman numerals. This system uses a set of characters to represent the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	numbers 1, 5, 10, 50, 100, 500, and 1000 and then those characters are combined with each other to construct the remaining values. The set of characters used by this numbering format is U+0049, U+0056, U+0058, U+004C, U+0043, U+0044, U+004D, respectively. To construct a number that is outside the set, you work from largest groups to smallest following these steps: 1. Create as many groups as possible that contain one thousand in each group. • The symbol representing one thousand (the power of ten represented by that position): M is repeated for the number of groups formed. If no groups are formed, do not write any symbol. 2. Repeat this for groups of nine hundred (CM), five-hundred (D), four-hundred (CD), one-hundred (C), ninety (XC), fifty (L), forty (XL), ten (X), nine (IX), five (V), four (IV) and finally one (I) using the corresponding symbol to indicate the groups (so four-hundred fifty would be CDL and forty-five would be XLV). [*Example*: The numbering for the items should be represented by the following pattern: I, II, III, IV, …, XVIII, XIX, XX, XXI, … *end example*]
vietnameseCounting (Vietnamese Numerals)	Specifies that the sequence shall consist of Vietnamese numerals. This sequence is a set of strings each of which is the full name, in Vietnamese, of the next value in that sequence. [*Example*: một, hai, ba, bốn, năm, sáu, bảy, tám, chín, mười. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NumberFormat) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.18.60 ST_ObjectDrawAspect (Embedded Object Representations)

This simple type specifies the ways in which embedded objects are displayed in the application.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
content (Snapshot)	The object's presentation is a picture of the contained document (provided by the object's server application).
icon (Icon)	The object's presentation is an icon.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ObjectDrawAspect) is located in §A.1. *end note*]

17.18.61 ST_ObjectUpdateMode (Embedded Object Update Modes)

This simple type specifies how an embedded object is updated.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
always (Server Application Update)	The object is updated whenever the object's server application indicates there is new data available.
onCall (User Update)	The object is updated when the user chooses to update it.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ObjectUpdateMode) is located in §A.1. *end note*]

17.18.62 ST_PageBorderDisplay (Page Border Display Options)

This simple type specifies the pages in the parent section on which the page border shall be printed.

[*Example*: Consider a section in a document for which the page border must only be printed on the first page. This setting is specified using the following WordprocessingML:

```
<w:pgBorders w:display="firstPage">
  …
</w:pgBorders>
```

The display attribute with a value of `firstPage` specifies that only the first page must display the page border defined for this section. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allPages (Display Page Border on All Pages)	Specifies that the page border shall be displayed on all pages in the parent section.
firstPage (Display Page Border on First Page)	Specifies that the page border shall be displayed on only the first page in the parent section.
notFirstPage (Display Page Border on All Pages Except First)	Specifies that the page border shall be displayed on only the first page in the parent section.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageBorderDisplay) is located in §A.1. *end note*]

17.18.63 ST_PageBorderOffset (Page Border Positioning Base)

This simple type specifies how the relative positioning of the page borders shall be calculated.

If the value of this attribute is `text`, then the space attribute on each page border shall be interpreted as the distance from the text margins that shall be left before the page border.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:pgBorders w:offsetFrom="page">
  <w:top w:val="dashed" w:space="24" />
  <w:left w:val="dashed" w:space="24" />
  <w:bottom w:val="dashed" w:space="24"/>
  <w:right w:val="dashed" w:space="24"/>
</w:pgBorders>
```

This fragment specifies that the page borders must be indented 24 points from the page extents.

This is distinct from the following fragment with identical space attribute values:

```
<w:pgBorders w:offsetFrom="text">
  <w:top w:val="dashed" w:space="24" />
  <w:left w:val="dashed" w:space="24" />
  <w:bottom w:val="dashed" w:space="24"/>
  <w:right w:val="dashed" w:space="24"/>
</w:pgBorders>
```

In this case, the page borders is offset by 24 points, but in this case, that offset is calculated relative to the text margins. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

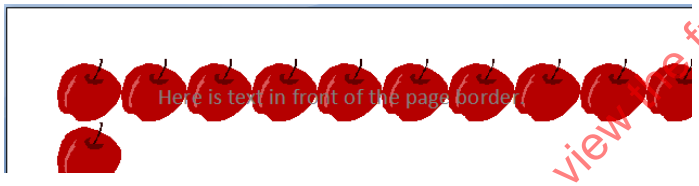
Enumeration Value	Description
page (Page Border Is Positioned Relative to Page Edges)	Specifies that the space attribute on each page border shall be interpreted as the distance from the edge of the page that shall be left before the page border.
text (Page Border Is Positioned Relative to Text Extents)	Specifies that the space attribute on each page border shall be interpreted as the distance from the edge of the text extents (text margins) that shall be left before the page border..

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageBorderOffset) is located in §A.1. *end note*]

17.18.64 ST_PageBorderZOrder (Page Border Z-Order)

This simple type specifies whether the page border is positioned above or below intersecting texts and objects in this document.

[*Example*: Consider a document in which the page border must be displayed below any intersecting text as follows:



This setting is specified by setting the value of an attribute with to back, which specifies that the page border must be displayed behind all intersecting text and objects. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
back (Page Border Behind Text)	Specifies that the page border shall be rendered beneath any text or object which intersects it - effectively placing it at the lowest z-order on the page.
front (Page Border Ahead of Text)	Specifies that the page border shall be rendered above any text or object which intersects it - effectively placing it at the highest z-order on the page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageBorderZOrder) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.18.65 ST_PageOrientation (Page Orientation)

This simple type specifies the orientation of all pages in the parent section. This information is used to determine the actual paper size to use when printing the file.

[*Example*: Pages 11" wide by 8.5" long in landscape mode use 8.5"x11" paper, because the width and height are reversed for pages in this landscape section with respect to the printed page. *end example*]

[*Example*: Consider the following WordprocessingML:

```
<w:pgSz w:w="15840" w:h="12240" w:orient="landscape" />
```

Although the page width is 11", and page height is 8.5", according to the w and h attributes, because the orient attribute is set to `landscape`, pages in this section are printed on 8.5x11" paper in landscape mode. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
landscape (Landscape Mode)	Specifies that pages in this section shall be printed in landscape mode, which prints the page contents with a 90 degree rotation with respect to the normal page orientation.
portrait (Portrait Mode)	Specifies that pages in this section shall be printed in portrait mode.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageOrientation) is located in §A.1. *end note*]

17.18.66 ST_Pitch (Font Pitch Value)

This simple type specifies the possible values for the font pitch of a font.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Courier New">
  <w:pitch w:val="fixed" />
  …
</w:font>
```

The pitch element specifies via its val attribute value of `fixed` that this is a fixed width font. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (Default)	Specifies that no information is available about the pitch of a font.
fixed (Fixed Width)	Specifies that this is a fixed width font.
variable (Proportional Width)	Specifies that this is a proportional width font.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Pitch) is located in §A.1. *end note*]

17.18.67 ST_PixelsMeasure (Measurement in Pixels)

This simple type specifies that its contents contain a positive whole number, whose contents consist of a measurement in pixels.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of 96 whose simple type is ST_PixelsMeasure. This attribute value specifies a size of 96 pixels (one inch on a 96 pixels per inch display). *end example*]

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber datatype (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PixelsMeasure) is located in §A.1. *end note*]

17.18.68 ST_PointMeasure (Measurement in Points)

This simple type specifies that its contents contain a positive whole number, whose contents consist of a measurement in points (equivalent to 1/72nd of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of 24 whose simple type is ST_PointMeasure. This attribute value specifies a size in points (24 points = 1/3 of an inch). *end example*]

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber datatype (§22.9.2.16).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PointMeasure) is located in §A.1. *end note*]

17.18.69 ST_Proof (Proofing State Values)

This simple type specifies the values which can be used to indicate the status of a given hosting application's grammar and spell checking when a given WordprocessingML document was last saved.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a WordprocessingML document that is saved by a hosting application whose grammar checking engine had completed checking the grammar in the given WordprocessingML document, but whose spell checking engine had not completed checking the spelling in the given WordprocessingML document . In this instance, the following WordprocessingML must be written in the document settings:

```
<w:proofState w:spelling="dirty" w:grammar="clean" />
```

The proofState element's attributes spelling and grammar have the ST_Proof simple type enumeration values `dirty` and `clean` respectively, specifying that the hosting application's spell checking engine had not completed checking the spelling of the document, and that the hosting application's grammar checking engine had completed checking the grammar of the document, when the WordprocessingML document was last saved. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clean (Check Completed)	Specifies that the given proofing engine completed checking the document when it was last saved.
dirty (Check Not Completed)	Specifies that the given proofing engine did not complete checking the document when it was last saved.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Proof) is located in §A.1. *end note*]

17.18.70 ST_ProofErr (Proofing Error Type)

This simple type specifies the possible values for the types of proofing error markers which can appear in the contents of a WordprocessingML document to indicate the last known state of any spell- and grammar-checking performed on the contents of this document.

[*Example*: Consider the following paragraph consisting of two misspelled words, where the second word has been explicitly flagged as not being a spelling error. This paragraph would consist of the following WordprocessingML markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:proofErr w:type="spellStart"/>
  <w:r>
    <w:t>erqwt</w:t>
  </w:r>
  <w:proofErr w:type="spellEnd"/>
  <w:r>
    <w:t xml:space="preserve"> werewr</w:t>
  </w:r>
</w:p>
```

The proofErr elements with a val attribute value of `spellStart` and `spellEnd`, respectively delimit the start and end the content in this paragraph which is stored as a spelling error. Since the second word is not included in that range, it is not stored as a spelling error. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
gramEnd (End of Region Marked as Grammatical Error)	Specifies that this proofing error marker shall indicate the start of a region to be marked as a grammatical error in the document.
gramStart (Start of Region Marked as Grammatical Error)	Specifies that this proofing error marker shall indicate the end of a region to be marked as a grammatical error in the document.
spellEnd (End of Region Marked as Spelling Error)	Specifies that this proofing error marker shall indicate the end of a region to be marked as a spelling error in the document.
spellStart (Start of Region Marked as Spelling Error)	Specifies that this proofing error marker shall indicate the start of a region to be marked as a spelling error in the document.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ProofErr) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.18.71 ST_PTabAlignment (Absolute Position Tab Alignment)

This simple type specifies the alignment of an absolutely positioned tab character in a document. This alignment value determines the position on the line to which this absolute tab shall advance, as well as the alignment of the text entered after the alignment tab character position.

[*Example*: Consider a positional tab stop in a WordprocessingML document who must move to the left edge of the text margins and whose subsequent text should be left aligned. This positional tab stop would be defined as follows:

 ©ISO/IEC 2016 – All rights reserved

```
<w:ptab w:alignment="left" w:relativeTo="margin" … />
```

The alignment attribute has a value of `left`, which specifies that this custom tab stop must align on the left edge of the line relative to the text margin. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center)	Specifies that the positional tab should be center aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be center aligned.
left (Left)	Specifies that the positional tab should be left aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be left aligned.
right (Right)	Specifies that the positional tab should be right aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be right aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PTabAlignment) is located in §A.1. *end note*]

17.18.72 ST_PTabLeader (Absolute Position Tab Leader Character)

This simple type specifies the characters which can be used to fill in the space created by a positional tab. This character shall be repeated as required to completely fill the tab spacing generated by the positional tab character.

[*Example*: Consider a positional tab stop which should be preceded by a sequence of underscore characters, as follows:

_____Text at the positional tab stop

This tab stop would have a leader attribute value of `underscore`, indicating that the tab stop must be preceded by underscore characters as needed to fill the tab spacing. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dot (Dot Leader Character)	Specifies that the leader character for this positional tab stop shall be a dot.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: ..... Text at absolute tab. *end example*]
hyphen (Hyphen Leader Character)	Specifies that the leader character for this positional tab stop shall be a hyphen. [*Example*: ---------- Text at absolute tab. *end example*]
middleDot (Centered Dot Leader Character)	Specifies that the leader character for this positional tab stop shall be a centered dot. [*Example*: ····· Text at absolute tab. *end example*]
none (No Leader Character)	Specifies that there shall be no leader character for this positional tab. [*Example*: Text at absolute tab. *end example*]
underscore (Underscore Leader Character)	Specifies that the leader character for this positional tab stop shall be an underscore. [*Example*: _____ Text at absolute tab. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PTabLeader) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

17.18.73 ST_PTabRelativeTo (Absolute Position Tab Positioning Base)

Specifies the possible extents which can be used to calculate the absolute positioning of this positional tab character.

[*Example*: Consider a positional tab stop in a WordprocessingML document that should have a resulting position that is centered on the text margins, ignoring both any custom tab stops and any text indents on the paragraph. This positional tab stop would be defined as follows:

```
<w:ptab w:relativeTo="margin" … />
```

The relativeTo attribute specifies that this absolute position tab stop must be relative to the margin. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
indent (Relative To Indents)	Specifies that the absolute positioning of the tab shall be relative to the indents.
margin (Relative To Text Margins)	Specifies that the absolute positioning of the tab shall be relative to the margins.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PTabRelativeTo) is located in §A.1. *end note*]

17.18.74 ST_RestartNumber (Footnote/Endnote Numbering Restart Locations)

This simple type specifies the possible values for when the automatic numbering of footnotes or endnotes shall be restarted.

[*Example*: Consider a WordprocessingML document where the numbering for its endnotes must be restarted after each section must be restarted after each page. This setting is represented by the following WordprocessingML:

```
<w:footnotePr>
  …
  <w:numRestart w:val="eachSect" />
  …
</w:footnotePr>
```

The val attribute value of eachSect specifies that numbering must be restarted after each section. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
continuous (Continue Numbering From Previous Section)	Specifies that the numbering of footnotes or endnotes shall continue from the previous section in the document.
eachPage (Restart Numbering On Each Page)	Specifies that the numbering of footnotes or endnotes shall be restarted to its starting value for each unique page in the document.
eachSect (Restart Numbering For Each Section)	Specifies that the numbering of footnotes or endnotes shall be restarted to its starting value for each unique section in the document.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RestartNumber) is located in §A.1. *end note*]

17.18.75 ST_RubyAlign (Phonetic Guide Text Alignment)

This simple type specifies the possible alignment settings which can be used to determine the placement of phonetic guide text with respect to the base text when this phonetic guide is displayed.

[*Example*: Consider a run of phonetic guide text which must have the ruby text positioned to the far left of the base text. This constraint is specified using the following WordprocessingML:

```
<w:rubyPr>
  …
  <w:rubyAlign w:val="left"/>
  …
</w:rubyPr>
```

The rubyAlign property is `left` for the phonetic guide, so the ruby text is displayed on the left side of the base text. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center)	Specifies that the phonetic guide text shall be centered with respect to the base text in this document. [*Example*: guide text this is a test center *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
distributeLetter (Distribute All Characters)	Specifies that the phonetic guide text shall be distributed with respect to the base text in this document. This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to all characters on the line. [*Example*: g u i d e t e x t a test distribute letter *end example*]
distributeSpace (Distribute all Characters w/ Additional Space On Either Side)	Specifies that the phonetic guide text shall be distributed with respect to the base text in this document, with additional space added to the guide text to ensure it is indented with respect to the base text. This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to all characters on the line. As well, an additional space is added before and after the guide text to ensure it is indented with respect to the base text. [*Example*: g u i d e t e x t a test distribute space *end example*]
left (Left Aligned)	Specifies that the phonetic guide text shall be left aligned with respect to the base text in this document. [*Example*: guide text this is a test left *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
right (Right Aligned)	Specifies that the phonetic guide text shall be right aligned with respect to the base text in this document. [*Example*: this is a test guide text right *end example*]
rightVertical (Vertically Aligned to Right of Base Text)	Specifies that the phonetic guide text shall be right aligned with respect to the base text in this document, and shall always be displayed vertically and to the right of the base text, regardless of the alignment of the base text. [*Example*: this is a test right vertical guide text *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RubyAlign) is located in §A.1. *end note*]

17.18.76 ST_SdtDateMappingType (Date Storage Format Types)

This simple type specifies then possible types of translations which can be performed on the displayed date in a date picker structured document tag when the current contents are saved into the associated custom XML data via the dataBinding element (§17.5.2.6).

[*Example*: Consider the following date picker structured document tag:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:sdt>
  <w:sdtPr>
    <w:date w:fullDate="01-01-2006T06:30:00Z">
      <w:storeMappedDateAs w:val="text"/>
      …
    </w:date>
  </w:sdtPr>
  <w:sdtContent>
    <w:r>
      <w:t>January 1</w:t>
    </w:r>
  </w:sdtContent>
</w:sdt>
```

The value of the storeMappedDateAs element's attribute value is `text`, therefore the current run contents must be sent to the mapped XML element without any translation (in this case, the value must be `January 1`). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
date (XML Schema Date Format)	Specifies that the date specified in the parent date picker structured document tag shall be converted to the xsd:date format when stored in a mapped XML element.
dateTime (XML Schema DateTime Format)	Specifies that the date specified in the parent date picker structured document tag shall be converted to the xsd:dateTime format when stored in a mapped XML element.
text (Same As Display)	Specifies that no translation shall be performed on the displayed date when stored in a mapped XML element - the mapped contents shall be the same as the displayed contents.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SdtDateMappingType) is located in §A.1. *end note*]

17.18.77 ST_SectionMark (Section Type)

Specifies the kind of the current section.

[*Example*: Consider a section that must start on the next page in the document. The WordprocessingML specifying this would look like:

```
<w:sectPr>
  …
  <w:type w:val="nextPage"/>
</w:sectPr>
```

The nextPage value specifies that this section starts on the next page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
continuous (Continuous Section Break)	Specifies a continuous section break, which begin the new section on the following paragraph. This means that continuous section breaks might not specify certain page-level section properties, since they are inherited from the following section. These breaks, however, can specify other section properties, such as line numbering and footnote/endnote settings. If a footnote reference (§17.11.14) occurs on the same page as a section break of this kind, the new section shall begin on the following page.
evenPage (Even Page Section Break)	Specifies an even page section break, which begins the new section on the next even-numbered page, leaving the next odd page blank if necessary.
nextColumn (Column Section Break)	Specifies a column section break, which begins the new section on the following column on the page.
nextPage (Next Page Section Break)	Specifies a next page section break, which begins the new section on the following page.
oddPage (Odd Page Section Break)	Specifies an odd page section break, which begins the new section on the next odd-numbered page, leaving the next even page blank if necessary.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SectionMark) is located in §A.1. *end note*]

[*Note*: A continuous section break balances the content of the previous section. *end note*]

[*Example*: Consider the following sample document:

©ISO/IEC 2016 – All rights reserved

Column-balancing-sample.....Section Break (Continuous).....

Text·text·text·

Column-balancing-sample Section Break (Continuous)

A¶
B¶

C¶
D¶

E Section Break (Continuous)

Last·section.¶

has incorrect column balancing, as the content is not laid out with respect to the implied constraints—although the section height is minimal. The correct minimal balancing looks like this:

Column-balancing-sample Section Break (Continuous)

A¶
B¶
C¶

D¶
E Section Break (Continuous)

Last·section.¶

end example]

[*Example*:

The following section balancing

Column-balancing-sample Section Break (Continuous)

A·long·long·long·long·long·long·long·long·
paragraph·with·Widow/Orphan·control·

turned·on. Section Break (Continuous)

Last·section.¶

is incorrect, as the “Widow/Orphan” constraints are not met. The correct balancing looks like this:

Column-balancing-sample Section Break (Continuous)

A·long·long·long·long·long·long·long·long·
paragraph·with·Widow/Orphan·control·
turned·on. Section Break (Continuous)

Last·section.¶

end example]

17.18.78 ST_Shd (Shading Patterns)

This simple type specifies the pattern that shall be used to lay the pattern color over the background color for shading.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This pattern consists of a mask that is applied over the background shading color to get the locations where the pattern color should be shown. Each of these possible masks is shown in the enumeration values below. In each example, an 8 pixel by 8 pixel mask is displayed where black has been used as the fill color (the parent element's fill attribute), and white has been used as the pattern color (the parent element's color attribute). When the shading is applied, the mask is tiled as necessary to match the size of the shaded area.

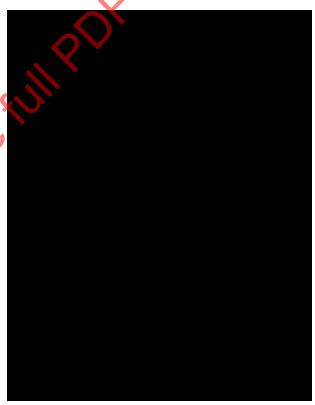
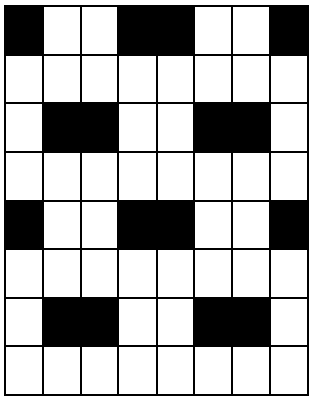
[*Example*: Consider a shaded paragraph which uses a 10 percent foreground fill, resulting in the following WordprocessingML:

```
<w:shd w:val="pct10" …/>
```

This shading val is `pct10`, indicating that the border style is a 10 percent foreground fill mask. *end example*]

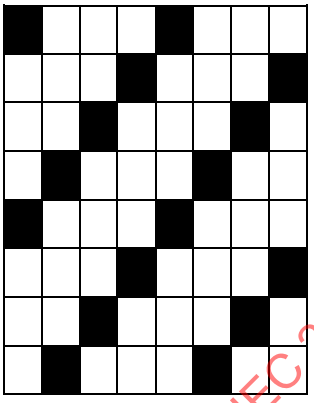
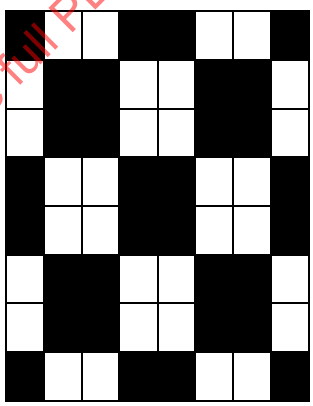
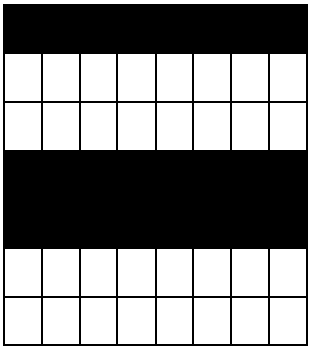
This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clear (No Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
diagCross (Diagonal Cross Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

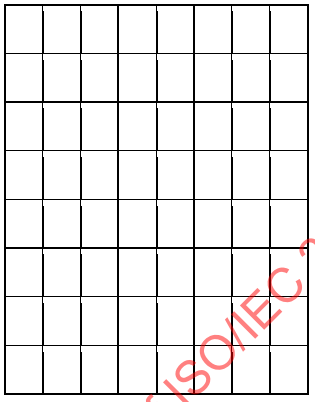
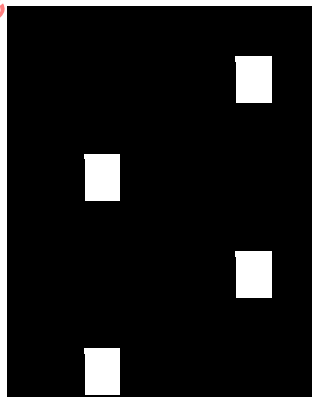
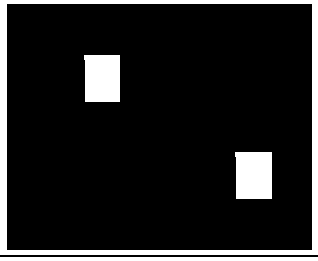
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
diagStripe (Diagonal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
horzCross (Horizontal Cross Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
horzStripe (Horizontal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

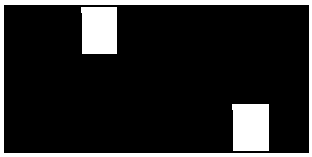
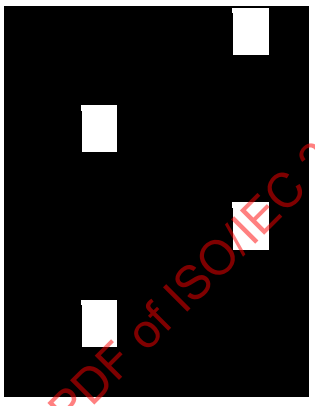
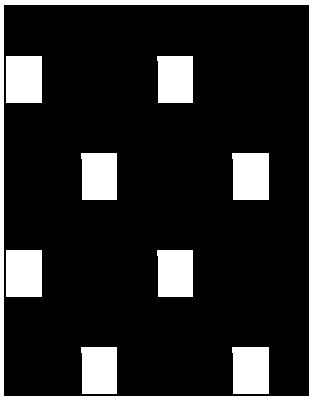
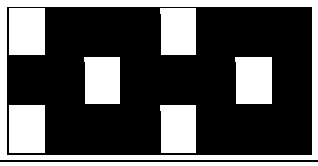
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
nil (No Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct10 (10% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct12 (12.5% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

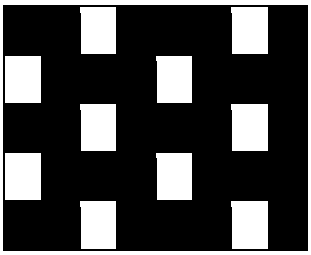
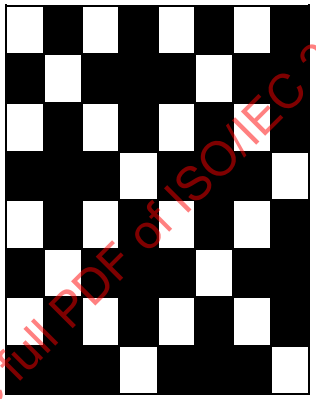
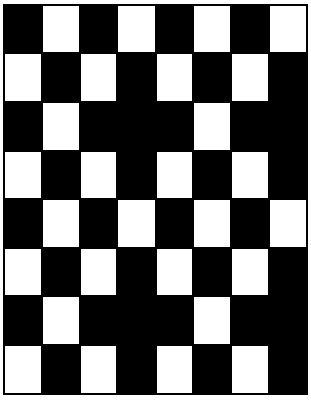
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct15 (15% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct20 (20% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct25 (25% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

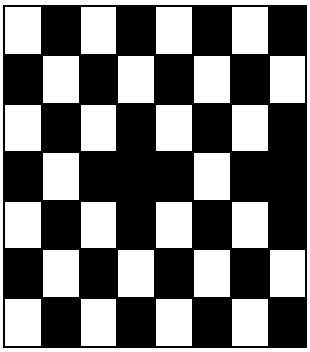
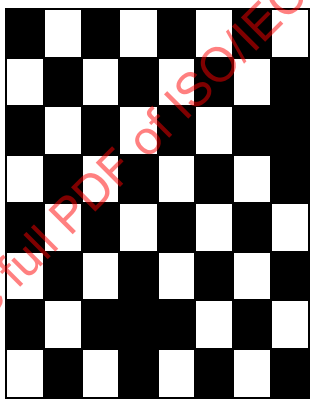
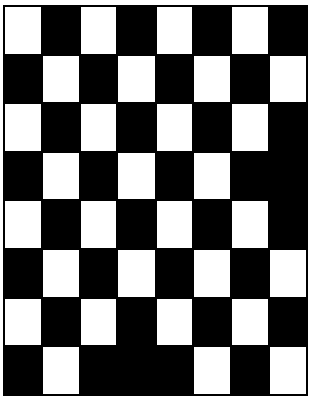
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct30 (30% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct35 (35% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct37 (37.5% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern:

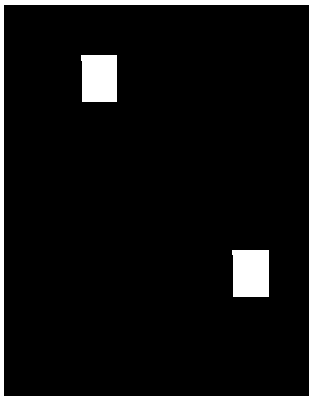
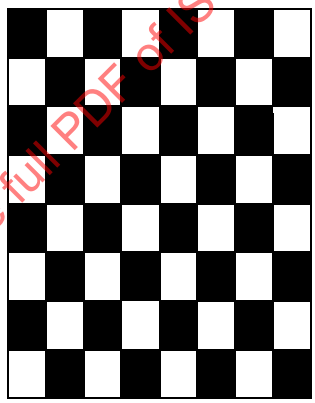
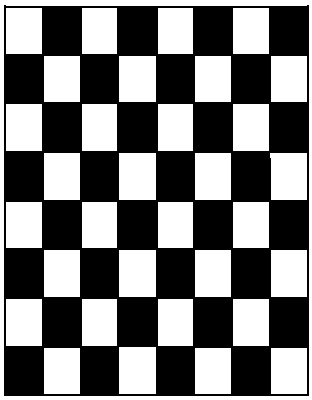
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct40 (40% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct45 (45% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct5 (5% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern:

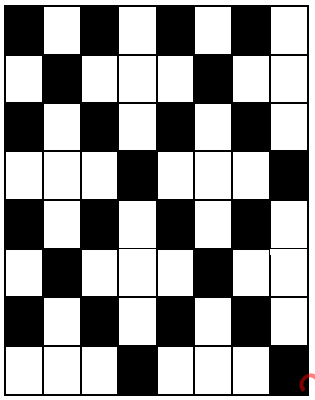
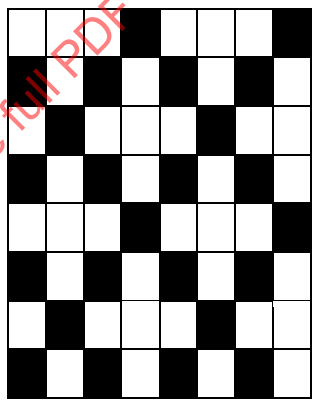
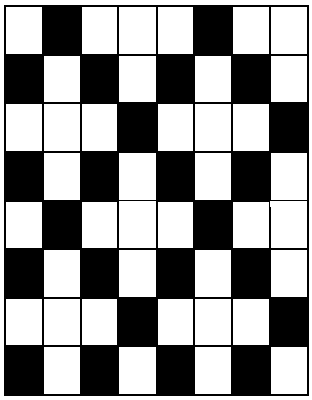
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct50 (50% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct55 (55% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

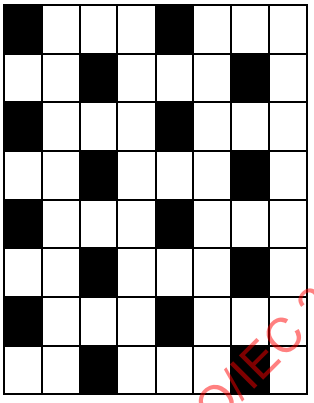
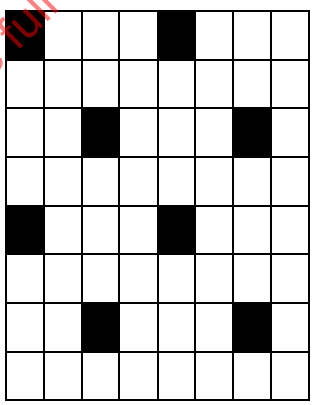
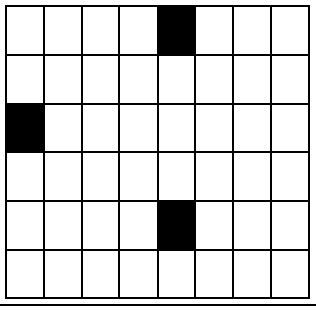
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
pct60 (60% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct62 (62.5% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct65 (65% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

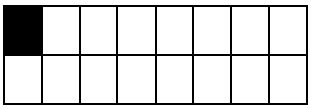
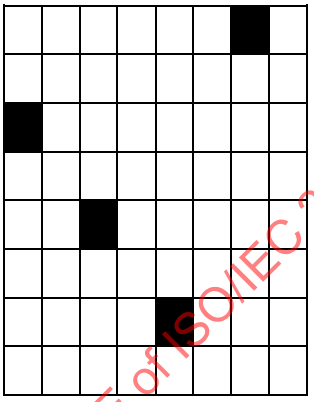
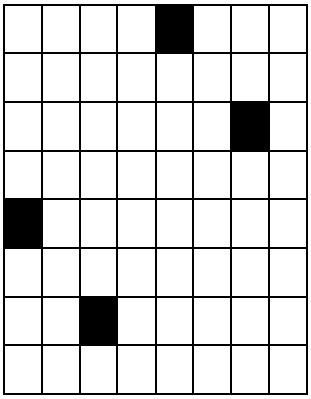
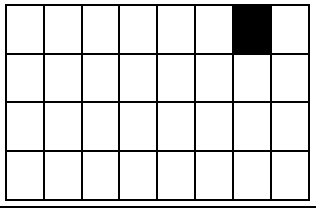
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
pct70 (70% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct75 (75% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct80 (80% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

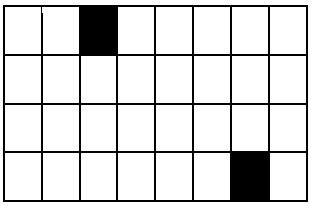
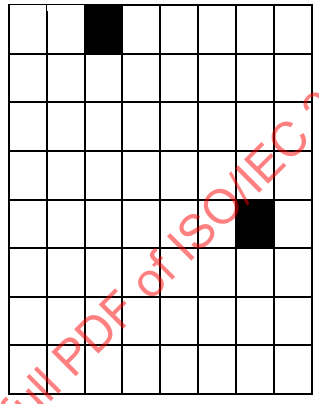
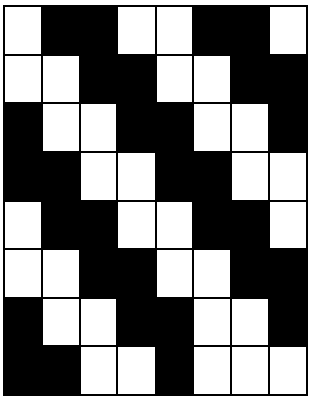
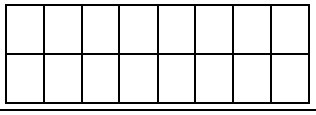
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct85 (85% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct87 (87.5% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
pct90 (90% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

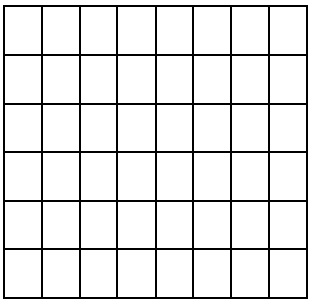
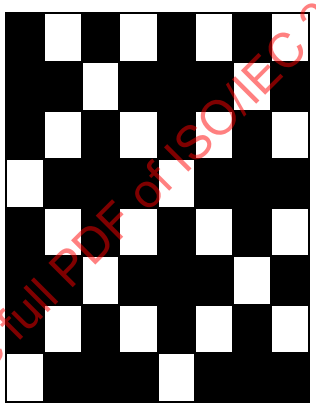
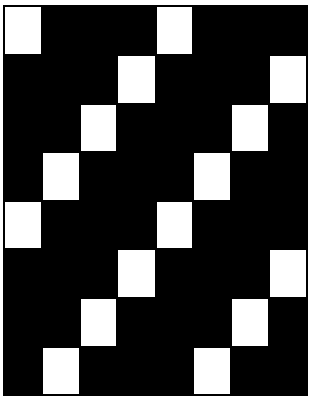
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
pct95 (95% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
reverseDiagStripe (Reverse Diagonal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
solid (100% Fill Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

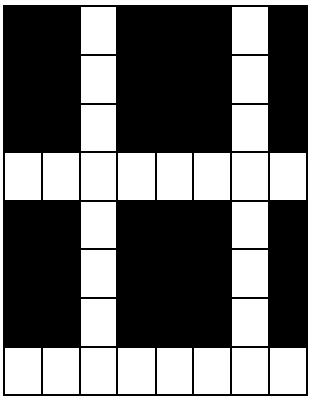
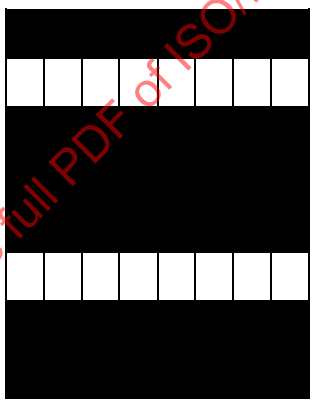
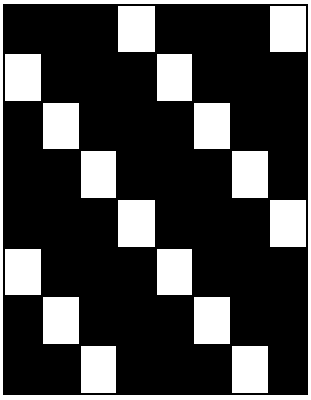
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
thinDiagCross (Thin Diagonal Cross Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
thinDiagStripe (Thin Diagonal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
thinHorzCross (Thin Horizontal Cross Pattern)	Specifies that the following mask shall be applied as the shading pattern:

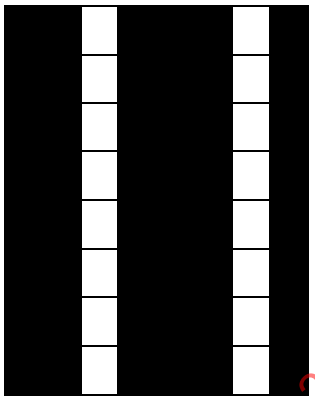
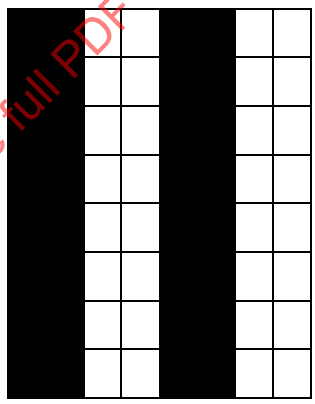
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	
thinHorzStripe (Thin Horizontal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
thinReverseDiagStripe (Thin Reverse Diagonal Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
thinVertStripe (Thin Vertical Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 
vertStripe (Vertical Stripe Pattern)	Specifies that the following mask shall be applied as the shading pattern: 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Shd) is located in §A.1. *end note*]

17.18.79 ST_ShortHexNumber (Four Digit Hexadecimal Value)

This simple type specifies a number value specified as a two octet (four digit) hexadecimal number, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following value for a node of type ST_ShortHexNumber: 2F6C.

This value is permitted, as it contains two hexadecimal octets, each an encoding of an octet of the actual decimal number value. *end example*]

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 4 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShortHexNumber) is located in §A.1. *end note*]

17.18.80 ST_SignedHpsMeasure (Signed Measurement in Half-Points)

This simple type specifies that its contents contain either:

- A positive or negative whole number, whose contents consist of a measurement in half-points (equivalent to 1/144th of an inch).
- A positive or negative number followed immediately by a unit identifier.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of `-72` whose type is ST_SignedHpsMeasure. This attribute value specifies a size of negative one-half of an inch or -36 points (`-72` halves of a point = -36 points = -0.5 inches). *end example*]

[*Example*: Consider an attribute value of `-12.7mm` whose type is ST_SignedHpsMeasure. This attribute value specifies a size of -0.0127 meter or negative one-half of an inch or -36 points. *end example*]

This simple type is a union of the following types:

- The ST_UniversalMeasure simple type (§22.9.2.15).
- The W3C XML Schema integer datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SignedHpsMeasure) is located in §A.1. *end note*]

17.18.81 ST_SignedTwipsMeasure (Signed Measurement in Twentieths of a Point)

This simple type specifies that its contents contain either:

- A positive or negative whole number, whose contents consist of a measurement in twentieths of a point (equivalent to 1/1440th of an inch), or
- A positive decimal number immediately following by a unit identifier.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of `720` whose simple type is ST_SignedTwipsMeasure. This attribute value specifies a size of one-half of an inch or 36 points (`720` twentieths of a point = 36 points = 0.5 inches). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider an attribute value of `-12.7mm` whose type is ST_SignedTwipsMeasure. This attribute value specifies a size of -0.0127 meter or negative one-half of an inch or -36 points. *end example*]

This simple type is a union of the following types:

- The ST_UniversalMeasure simple type (§22.9.2.15).
- The W3C XML Schema integer datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SignedTwipsMeasure) is located in §A.1. *end note*]

17.18.82 ST_StyleSort (Style Sort Settings)

This simple type specifies the ways in which the list of document styles can be ordered when they are displayed in a user interface.

[*Example*: Consider the WordprocessingML below:

```
<w:stylePaneSortMethod w:val="basedOn" />
```

The stylePaneSortMethod element has a val attribute value of `basedOn`, which specifies that styles are sorted by the value of their basedOn element (§**17.7.4.3**). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
basedOn (Sort by Based On Style)	Specifies that styles which are visible should be sorted by the style on which they are based using the basedOn element (§**17.7.4.3**).
default (Sort by Default Method)	Specifies that styles which are visible should be sorted by the default sorting of the host application.
font (Sort by Font)	Specifies that styles which are visible should be sorted by the font which they apply.
name (Sort by Style Name)	Specifies that styles which are visible should be sorted by their names.
priority (Sort by Style Priority)	Specifies that styles which are visible should be sorted by their UI priority using the uiPriority element (§**17.7.4.19**).
type (Sort by Style Type)	Specifies that styles which are visible should be sorted by their style types (i.e. character, linked, paragraph).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StyleSort) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

17.18.83 ST_StyleType (Style Types)

This simple type specifies the possible values for the types of style definitions defined within a WordprocessingML document. WordprocessingML supports six types of style definitions:

- Paragraph styles
- Character styles
- Table styles
- Numbering styles
- Linked styles (paragraph + character)
- Default paragraph + character properties

Each of the first four style types corresponds to a different value below, and therefore defines the style type of the current style. [*Note*: The last two style types are unique in that they are not simply a style type: a linked style is a pairing of a character and paragraph style via the link element (§17.7.4.6); and the document default properties are defined via the docDefaults element (§17.7.5.1). *end note*]

[*Example*: Consider a style defined as follows:

```
<w:style w:type="paragraph" … >
  <w:name w:val="My Paragraph Style"/>
  <w:rPr>
    <w:b/>
  </w:rPr>
</w:style>
```

The type attribute is of simple type ST_StyleType, and its value of paragraph specifies that this style definition creates a paragraph style. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
character (Character Style)	Specifies that the parent style definition is a character style.
numbering (Numbering Style)	Specifies that the parent style definition is a numbering style.
paragraph (Paragraph Style)	Specifies that the parent style definition is a paragraph style.
table (Table Style)	Specifies that the parent style definition is a table style.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StyleType) is located in §A.1. *end note*]

17.18.84 ST_TabJc (Custom Tab Stop Type)

This simple type specifies the available types of custom tab stop, which determines the behavior of the tab stop and the alignment which shall be applied to text entered at the current custom tab stop.

[*Example*: Consider a custom tab stops at 1.5" in a WordprocessingML document. This tab stop would be contained within a tab element defining the tab stop as follows:

```
<w:tab w:val="start" w:pos="2160" />
```

The val attribute specifies that this custom tab stop must align all text entered at its location to its left. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar Tab)	Specifies that the current tab is a bar tab. A *bar tab* is a tab which does not result in a custom tab stop in the parent paragraph (this tab stop location shall be skipped when positioning custom tab characters), but instead shall be used to draw a vertical line (or bar) at this location in the parent paragraph.
center (Centered Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is centered (i.e. all text runs following this tab stop and preceding the next tab stop shall be centered around the tab stop location).
clear (No Tab Stop)	Specifies that the current tab stop is cleared and shall be removed and ignored when processing the contents of this document.
decimal (Decimal Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is aligned around the first decimal character in the following text runs. All text runs before the first decimal character shall be before the tab stop, all text runs after it shall be after the tab stop location.
end (Trailing Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is aligned to its trailing edge (i.e. all text runs following this tab stop and preceding the next tab stop shall be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	aligned against the trailing edge with respect to the tab stop location). [*Example*: In an RTL paragraph, the trailing edge is the left edge, so text aligns to that edge, extending to the right. *end example*]
num (List Tab)	Specifies that the current tab is a list tab, which is the tab stop between the numbering and the paragraph contents in a numbered paragraph. [*Note*: This justification style is used for backwards compatibility with earlier word processors, and should be avoided in favor of hanging paragraph indentation. *end note*]
start (Leading Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is aligned to its leading edge (i.e. all text runs following this tab stop and preceding the next tab stop shall be aligned against the leading edge with respect to the tab stop location).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TabJc) is located in §A.1. *end note*]

17.18.85 ST_TabTlc (Custom Tab Stop Leader Character)

This simple type specifies the characters which can be used to fill in the space created by a tab which ends at this custom tab stop. The chosen character shall be repeated as required to completely fill the tab spacing generated by the tab character.

[*Example*: Consider a tab stop which should be preceded by a sequence of underscore characters, as follows:

_____Text at the tab stop

This tab stop would have a leader attribute value of `underscore`, indicating that the tab stop must be preceded by underscore characters as needed to fill the tab spacing. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dot (Dotted leader line)	Specifies that the leader character for this custom tab stop shall be a dot. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	.....Text at tab stop. *end example*]
heavy (Heavy solid leader line)	Specifies that the leader character for this custom tab stop shall be a heavy solid line, or an underscore. [*Note*: This setting is used for backwards compatibility with earlier word processors, and should be avoided in favor of other leader characters. It can be displayed using underscores if desired. *end note*] [*Example*: _____Text at tab stop. *end example*]
hyphen (Dashed tab stop leader line)	Specifies that the leader character for this custom tab stop shall be a hyphen. [*Example*: ----------Text at tab stop. *end example*]
middleDot (Middle dot leader line)	Specifies that the leader character for this custom tab stop shall be a centered dot. [*Example*: ·····Text at tab stop. *end example*]
none (No tab stop leader)	Specifies that there shall be no leader character for this custom tab. [*Example*: Text at tab stop. *end example*]
underscore (Solid leader line)	Specifies that the leader character for this custom tab stop shall be an underscore. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	_____Text at tab stop. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TabTlc) is located in §A.1. *end note*]

17.18.86 ST_TargetScreenSz (Target Screen Sizes for Generated Web Pages)

This simple type specifies possible ideal minimum target screen sizes (width by height, specified in pixels) for which web pages generated can be optimized when saving this document as a web page.

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

```
<w:webSettings>
  <w:targetScreenSz w:val="1600x1200" />
</w:webSettings>
```

The targetScreenSz element's val attribute has a value of 1600x1200, which specifies that a target screen size of 1600 by 1200 pixels must be assumed when saving this document as a web page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1024x768 (Optimize for 1024x768)	Specifies that web pages produced from this document should be optimized for a screen size of 1024x768.
1152x882 (Optimize for 1152x882)	Specifies that web pages produced from this document should be optimized for a screen size of 1152x882.
1152x900 (Optimize for 1152x900)	Specifies that web pages produced from this document should be optimized for a screen size of 1152x900.
1280x1024 (Optimize for 1280x1024)	Specifies that web pages produced from this document should be optimized for a screen size of 1280x1024.
1600x1200 (Optimize for 1600x1200)	Specifies that web pages produced from this document should be optimized for a screen size of 1600x1200.
1800x1440 (Optimize for 1800x1440)	Specifies that web pages produced from this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	document should be optimized for a screen size of 1800x1440.
1920x1200 (Optimize for 1920x1200)	Specifies that web pages produced from this document should be optimized for a screen size of 1920x1200.
544x376 (Optimize for 544x376)	Specifies that web pages produced from this document should be optimized for a screen size of 544x376.
640x480 (Optimize for 640x480)	Specifies that web pages produced from this document should be optimized for a screen size of 640x480.
720x512 (Optimize for 720x512)	Specifies that web pages produced from this document should be optimized for a screen size of 720x512.
800x600 (Optimize for 800x600)	Specifies that web pages produced from this document should be optimized for a screen size of 800x600.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TargetScreenSz) is located in §A.1. *end note*]

17.18.87 ST_TblLayoutType (Table Layout Type)

This simple type defines the possible types of layout algorithms which can be used to lay out a table within a WordprocessingML document.

These algorithms are defined in the following paragraphs (noting, of course, that implementations are free to implement more efficient versions of each).

Fixed Width Table Layout - This method of table layout uses the preferred widths on the table items to generate the final sizing of the table, but does not change that size regardless of the contents of each table cell, hence the table is fixed width.

[*Guidance*: Although an application can choose to use a different process, this layout could be performed as follows:

- The table grid is used to create the set of shared columns in the table and their initial widths as defined in the tblGrid element (§17.4.48)
- The table's total width is defined based on the tblW property (§17.4.63) – if it is set to `auto` or `nil`, then the width is not yet determined and is specified using the row and cell information.
- The first table row is read and the initial number of grid units before the row starts is skipped. The width of the skipped grid columns is set using the wBefore property (§17.4.86).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- The first cell is placed on the grid, and the width of the specified grid column span set by gridSpan (§17.4.17) is set based on the tcW property (§17.4.71).
- Each additional cell is placed on the grid.
- If at any stage, the preferred width requested for the cells exceeds the preferred width of the table, then each grid column is proportionally reduced in size to fit the table width.
- If the grid is exceeded (e.g. tblGrid specifies three grid columns, but the second cell has a gridSpan of three), the grid is dynamically increased with a default width for the new grid column.
- For each subsequent row, cells are placed on the grid, and each grid column is adjusted to be the maximum value of the requested widths (if the widths do not agree) by adding width to the last cell that ends with that grid column. Again, if at any point, the space requested for the cells exceeds the width of the table, then each grid column is proportionally reduced in size to fit the table width.

end guidance]

The resulting table shall be displayed regardless of its contents to the size requested.

AutoFit Table Layout - This method of table layout uses the preferred widths on the table items to generate the final sizing of the table, but then uses the contents of each cell to determine final column widths.

[*Guidance*: This layout can be performed in any manner available to an application, but one algorithm as follows can be used:

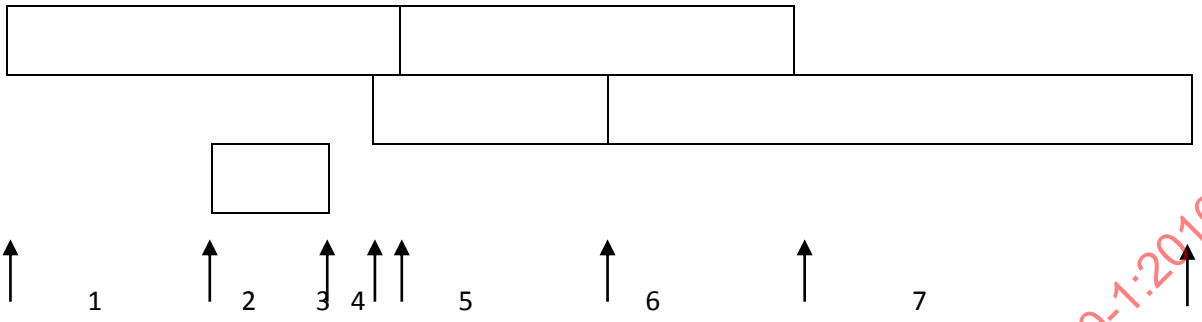
- Perform the steps above to lay out the fixed width version of the table.
- Calculate the minimum content width - the width of the cell's contents including all possible line breaking locations (or the cell's width, if the width of the content is smaller), and the maximum content width -the width of the cell's contents (assuming no line breaking not generated by explicit line breaks).
- The minimum and maximum content width of all cells that span a single grid column is the minimum and maximum content width of that column.
- For cells which span multiple grid columns, enlarge all cells which it spans as needed to meet that cell's minimum width.
- If any cell in a grid column has a preferred width, the first such width overrides the maximum width of the column's contents.
- Place the text in the cells in the table, respecting the minimum content width of each cell's content. If a cell's minimum content width exceeds the cell's current width, preferences are overridden as follows:
- First, override the column widths by making all other grid columns proportionally smaller until each it at its minimum width. This cell can then grow to any width between its own minimum and maximum width.
- Next, override the preferred table width until the table reaches the page width.
- Finally, force a line break in each cell's contents as needed

end guidance]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following fixed width table, which makes extensive use of resized and merged cells on what is actually just a seven-column grid. (The arrows point to each (invisible) vertical line of the grid and the numbers refer to the grid columns):



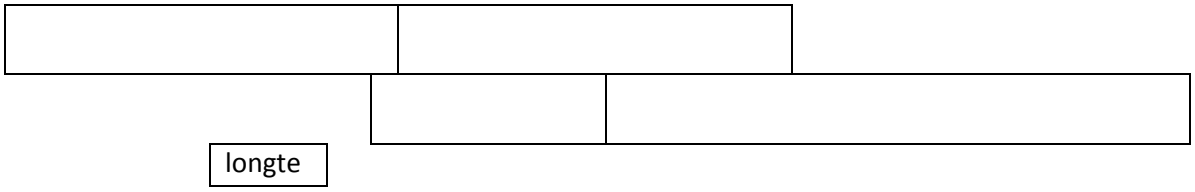
Although the table is visually complex, the standard rules apply: the first cell in the table is simply a cell which spans four grid units horizontally, as specified in the gridSpan element, and whose preferred with is 2952 twentieths of a point, specified in the tcW element:

```
<w:tc>
  <w:tcPr>
    <w:tcW w:w="2952" w:type="dxa"/>
    <w:gridSpan w:val="4"/>
  </w:tcPr>
  <w:p/>
</w:tc>
```

Similarly, all cells indented from the stand and end of the grid specify that indent using the gridBefore and gridAfter elements. For example, the XML for the second row in the table shows that that row starts three grid units into the table:

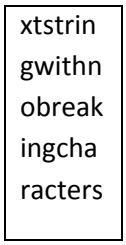
```
<w:tr>
  <w:trPr>
    <w:gridBefore w:val="3"/>
    <w:wBefore w:w="2748" w:type="dxa"/>
  </w:trPr>
  …
</w:tr>
```

If we take this fixed width table and introduce a long string into the single cell in row three, we see that the presence of this text does not affect cell widths:

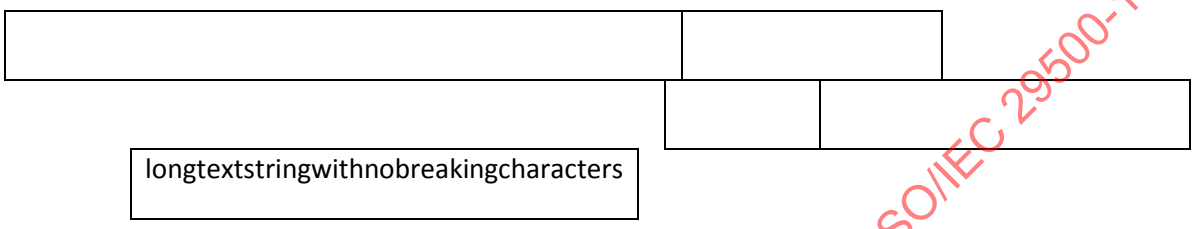


IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



If we now turn on the AutoFit property, we see that the algorithm for this AutoFit table causes grid column two to increase in size, proportionally decreasing the other grid columns' size to accommodate the long non-breaking string in the last cell:



Each of the other grid columns was reduced, but since all columns are not at their minimum size, the table width is not increased even though the table is not yet at the page width. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autofit (AutoFit Table Layout)	Specifies that this table shall use an AutoFit table layout algorithm.
fixed (Fixed Width Table Layout)	Specifies that this table shall use the fixed width table layout algorithm described above.

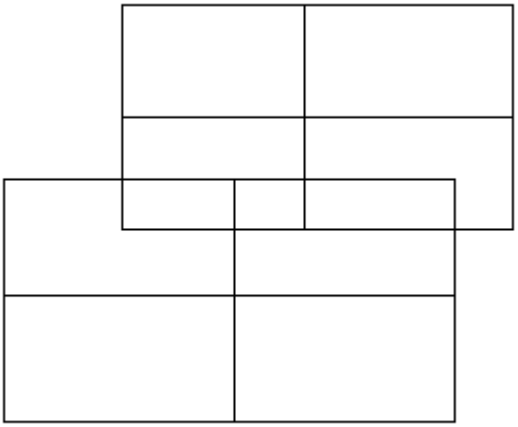
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TblLayoutType) is located in §A.1. *end note*]

17.18.88 ST_TblOverlap (Table Overlap Setting)

This simple type contains the possible settings for a floating table which shall be used to determine if the table can overlap with other floating tables when displayed in the document.

[*Example*: Consider two floating tables in a WordprocessingML document which overlap when displayed, as follows:

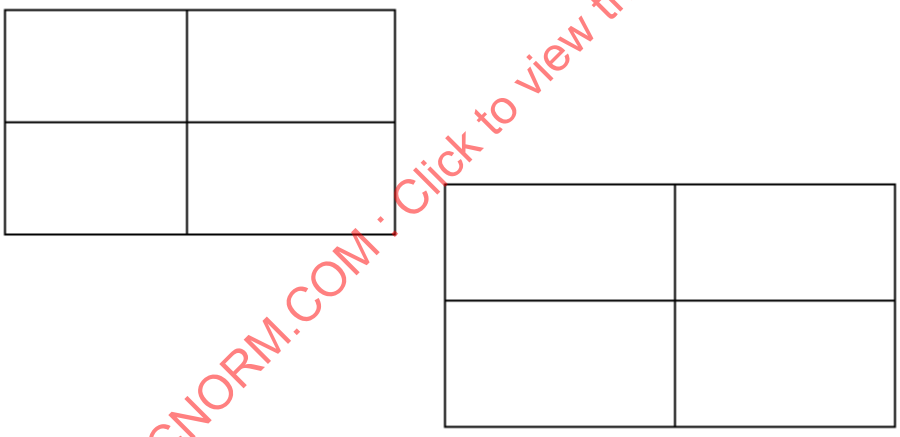
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



If either of these tables specifies that it must not allow overlapping, using the following WordprocessingML:

```
<w:tblPr>
  <w:tblOverlap w:val="never"/>
</w:tblPr>
```

The resulting tables must not overlap, and shall be adjusted at display time to prevent any overlapping, for example:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The value of never specifies that the specified table cannot overlap with other floating tables in the document. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
never (Floating Table Cannot Overlap)	Specifies that the parent table, if floating, shall never be displayed in a state where it would be overlapping another floating table in the document. If two floating tables intersect and this option is set on either of them, then one or both tables shall be adjusted as needed to ensure that the table whose value is never is not overlapped when displayed.
overlap (Floating Table Can Overlap)	Specifies that the parent table, if floating, can be displayed in a state where it would be overlapping another floating table in the document.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TblOverlap) is located in §A.1. *end note*]

17.18.89 ST_TblStyleOverrideType (Conditional Table Style Formatting Types)

This simple type specifies possible values for the sections of the table to which the current conditional formatting properties shall be applied when this table style is used.

[*Example*: Consider a table style which contains conditional formatting, defined as follows:

```
<w:style w:type="table" …>
  …
  <w:tblStylePr w:type="lastRow">
    …
  </w:tblStylePr>
</w:style>
```

The type attribute value of `lastRow` specifies that this set of conditional formatting properties must be applied to the last row of the table only. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
band1Horz (Banded Row Conditional Formatting)	Specifies that the table formatting applies to odd numbered groupings of rows.
band1Vert (Banded Column Conditional Formatting)	Specifies that the table formatting applies to odd numbered groupings of columns.
band2Horz (Even Row Stripe Conditional Formatting)	Specifies that the table formatting applies to even numbered groupings of rows.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
band2Vert (Even Column Stripe Conditional Formatting)	Specifies that the table formatting applies to even numbered groupings of columns.
firstCol (First Column Conditional Formatting)	Specifies that the table formatting applies to the first column.
firstRow (First Row Conditional Formatting)	Specifies that the table formatting applies to the first row. Any subsequent row which has the tblHeader element present (§17.4.49) shall also use this conditional format.
lastCol (Last table column formatting)	Specifies that the table formatting applies to the last column.
lastRow (Last table row formatting)	Specifies that the table formatting applies to the last row.
neCell (Top right table cell formatting)	Specifies that the table formatting applies to the top right cell.
nwCell (Top left table cell formatting)	Specifies that the table formatting applies to the top left cell.
seCell (Bottom right table cell formatting)	Specifies that the table formatting applies to the bottom right cell.
swCell (Bottom left table cell formatting)	Specifies that the table formatting applies to the bottom left cell.
wholeTable (Whole table formatting)	Specifies that the conditional formatting applies to the whole table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TblStyleOverrideType) is located in §A.1. *end note*]

17.18.90 ST_TblWidth (Table Width Units)

This simple type specifies the possible values for the units of the width property being defined by a specific table width property. These properties are used to define various properties of a table, including: cell spacing, preferred width, and table margins.

[*Example*: Consider a table with a table cell bottom cell spacing with a type of dxa, as follows:

```
<w:bottom … w:type="dxa" />
```

This type must therefore be used to interpret the width specified in the w attribute as a value in twentieths of a point. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatically Determined Width)	Specifies that the value for the measurement of the current table width property in the parent table shall be automatically determined by the table layout algorithm when the table is displayed (this width can be adjusted as appropriate). If this value is inappropriate for the current measurement (i.e. this measurement is not affected by that algorithm), then this width type and the associated value can be ignored.
dxa (Width in Twentieths of a Point)	Specifies that the value for the measurement of the current table width property in the parent table shall be interpreted as twentieths of a point (1/1440 of an inch).
nil (No Width)	Specifies that the current width is zero, regardless of any width value specified on the parent element.
pct (Width in Percent of Table Width)	Specifies that the value for the measurement of the current table width property in the parent table shall be interpreted as whole percentage point when a percent sign (U+0025) is present. These percentages shall be calculated relative to the extents specified by the parent XML element. If this value is inappropriate for the current measurement (i.e. this measurement is not part of the width of the table), then this width type and the associated value can be ignored.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TblWidth) is located in §A.1. *end note*]

17.18.91 ST_TextAlignment (Vertical Text Alignment Types)

This simple type specifies the type of vertical alignment which shall be used to align the characters on each line in the parent object.

[*Example*: Consider a paragraph of text of different font sizes, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If the text on this paragraph must be aligned based on the top point of the maximum character height, that requirement would be specified as follows in the WordprocessingML:

```
<w:pPr>
  <w:textAlignment w:val="top" />
</w:pPr>
```

The resulting text would be top aligned, as follows:

The characters are all aligned to the maximum character extent on the line. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatically Determine Alignment)	Specifies that all text in the parent object shall be aligned automatically when displayed.
baseline (Align Text at Baseline)	Specifies that all text in the parent object shall be aligned to the baseline of each character when displayed.
bottom (Align Text at Bottom)	Specifies that all text in the parent object shall be aligned to the bottom of each character when displayed.
center (Align Text at Center)	Specifies that all text in the parent object shall be aligned to the center of each character when displayed.
top (Align Text at Top)	Specifies that all text in the parent object shall be aligned to the top of each character when displayed.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAlignment) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

17.18.92 ST_TextboxTightWrap (Lines To Tight Wrap Within Text Box)

This simple type specifies the lines in the parent paragraph which shall allow the text to be tight wrapped to the paragraph (and not the containing text box) extents when displaying the document.

[*Example*: Consider a paragraph in a text box which meets the criteria specified above which must allow wrapping to the text extents on its first line only. That requirement would be specified using the following WordprocessingML:

 ©ISO/IEC 2016 – All rights reserved

```
<w:pPr>
  <w:textboxTightWrap w:val="firstLineOnly" />
</w:pPr>
```

The resulting paragraph would allow text to tightly wrap to the contents of its first line only. All other lines would wrap to the text box's extents. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allLines (Tight Wrap All Lines)	Specifies that all lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
firstAndLastLine (Tight Wrap First and Last Lines)	Specifies that only the first and last lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
firstLineOnly (Tight Wrap First Line)	Specifies that only the first line in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
lastLineOnly (Tight Wrap Last Line)	Specifies that only the last line in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
none (Do Not Tight Wrap)	Specifies that no lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextboxTightWrap) is located in §A.1. *end note*]

17.18.93 ST_TextDirection (Text Flow Direction)

This simple type specifies the direction of the text flow for the parent object.

[*Example*: Consider an object in which text must be oriented vertically, flowing from left to right horizontally on the page. This is achieved by using an `lr` value in an element of type ST_TextDirection. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lr (Lines Flow From Left to Right)	Specifies that text in the parent object shall be oriented vertically, flowing from left to right

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	horizontally on the page. This means that vertical lines are filled before the text expands horizontally.
lrV (Lines Flow From Left to Right Rotated)	Specifies that text in the parent object shall be oriented vertically, flowing from left to right horizontally on the page. This means that vertical lines are filled before the text expands horizontally. This flow is also rotated clockwise such that text which is not in an East Asian script is rotated 90 degrees when displayed on a page.
rl (Lines Flow From Right to Left)	Specifies that text in the parent object shall be oriented vertically, flowing from right to left horizontally on the page, as if the text were rotated 90 degrees. This means that vertical lines are filled before the text expands horizontally.
rlV (Lines Flow From Right to Left Rotated)	Specifies that text in the parent object shall be oriented vertically, flowing from right to left horizontally on the page. This means that vertical lines are filled before the text expands horizontally. This flow is also rotated such that text which is not in an East Asian script is rotated 90 degrees clockwise when displayed on a page.
tb (Lines Flow From Top to Bottom)	Specifies that text in the parent object shall be oriented horizontally, flowing from top to bottom vertically on the page. The text direction can be set to right-to-left using the bidi element (§17.3.1.6) within individual paragraphs. This means that horizontal lines are filled before the text expands vertically.
tbV (Lines Flow From Top to Bottom Rotated)	Specifies that text in the parent object shall be oriented horizontally, flowing from top to bottom vertically on the page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	This means that horizontal lines are filled before the text expands vertically. This flow is also rotated such that any characters from an East Asian script shall be rotated 270 degrees clockwise when displayed on a page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextDirection) is located in §A.1. *end note*]

17.18.94 ST_TextEffect (Animated Text Effects)

This simple type specifies the possible types of animated text effect which can be applied to a text run when it is displayed..

[*Example*: Consider a run of text which must have an animated text effect consisting of multiple colored flashing lights. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:effect w:val="lights"/>
</w:rPr>
```

This run explicitly declares a type of text effect, using the val property, of `lights`, so the contents of this run has the animated lights text effect. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
antsBlack (Black Dashed Line Animation)	Specifies that this text shall be surrounded by an animated black dashed line border.
antsRed (Marching Red Ants)	Specifies that this text shall be surrounded by an animated red dashed line border.
blinkBackground (Blinking Background Animation)	Specifies that this text shall be surrounded by a background color which alternates between black and white.
lights (Colored Lights Animation)	Specifies that this text shall be surrounded by a border consisting of a series of colored lights, which constantly change colors in sequence.
none (No Animation)	Specifies that this text shall have no animated text effect.
shimmer (Shimmer Animation)	Specifies that this text shall be animated by alternating between normal and blurry states.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
sparkle (Sparkling Lights Animation)	Specifies that this text shall have a background consisting of a random pattern of colored lights, which constantly change colors in sequence.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextEffect) is located in §A.1. *end note*]

17.18.95 ST_TextScale (Text Expansion/Compression Percentage)

This simple type specifies that the percentage by which the contents of a run shall be expanded or compressed with respect to its normal (100%) character width, with a minimum width of 1% and maximum width of 600%.

[*Example*: Consider a run of text which must be compressed by half when displaying each character within the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:w w:val="50%"/>
</w:rPr>
```

This run explicitly declares that the w value is 50%, so the contents of this run appear at 50% of their normal character width by compressing the width of each character. *end example*]

This simple type is a union of the following types:

- The ST_TextScalePercent simple type (§A.1).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextScale) is located in §A.1. *end note*]

17.18.96 ST_Theme (Theme Font)

This simple type specifies a theme font type which can be referenced as a theme font within the parent run properties. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part,which allows for font information to be set centrally in the document.

[*Example*: Consider a run of text which must be displayed using the `majorASCII` theme font. This requirement would be specified as follows in the resulting WordprocessingML:

```
<w:rPr>
  <w:rFonts w:asciiTheme="majorAscii" />
</w:rPr>
```

The ascii attribute specifies that the run must use the `majorAscii` theme font as defined in the document's themes part for all text in the appropriate range. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
majorAscii (Major ASCII Theme Font)	Specifies that the current font is a reference to the major theme font for the range of characters from U+0000–U+007F.
majorBidi (Major Complex Script Theme Font)	Specifies that the current font is a reference to the major theme font for the Complex Script range.
majorEastAsia (Major East Asian Theme Font)	Specifies that the current font is a reference to the major theme font for the East Asian range.
majorHAnsi (Major High ANSI Theme Font)	Specifies that the current font is a reference to the major theme font for the High ANSI range.
minorAscii (Minor ASCII Theme Font)	Specifies that the current font is a reference to the minor theme font for the range of characters from U+0000–U+007F.
minorBidi (Minor Complex Script Theme Font)	Specifies that the current font is a reference to the minor theme font for the Complex Script range.
minorEastAsia (Minor East Asian Theme Font)	Specifies that the current font is a reference to the minor theme font for the East Asian range.
minorHAnsi (Minor High ANSI Theme Font)	Specifies that the current font is a reference to the minor theme font for the High ANSI range.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Theme) is located in §A.1. *end note*]

17.18.97 ST_ThemeColor (Theme Color)

This simple type specifies a theme color to be applied to the current object. The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows color information to be set centrally in the document.

[*Example*: Consider a set of borders configured to use the `accent2` theme color, resulting in the following WordprocessingML markup:

```
<w:top … w:color="FFA8A0" w:themeColor="accent2" w:themeTint="99" />
<w:bottom … w:color="FFA8A0" w:themeColor="accent2" w:themeTint="99" />
<w:left … w:color="FFA8A0" w:themeColor="accent2" w:themeTint="99" />
<w:right … w:color="FFA8A0" w:themeColor="accent2" w:themeTint="99" />
```

The borders have a themeColor attribute of type ST_ThemeColor that when specified, imports the `accent2` theme color specified for this document. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent 1 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent1 attribute of the clrSchemeMapping element (§17.15.1.20).
accent2 (Accent 2 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent2 attribute of the clrSchemeMapping element (§17.15.1.20).
accent3 (Accent 3 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent3 attribute of the clrSchemeMapping element (§17.15.1.20).
accent4 (Accent 4 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent4 attribute of the clrSchemeMapping element (§17.15.1.20).
accent5 (Accent 5 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent5 attribute of the clrSchemeMapping element (§17.15.1.20).
accent6 (Accent 6 Theme Color)	Specifies that the color to be used shall be the theme color specified by the accent6 attribute of the clrSchemeMapping element (§17.15.1.20).
background1 (Background 1 Theme Color)	Specifies that the color to be used shall be the theme color specified by the bg1 attribute of the clrSchemeMapping element (§17.15.1.20).
background2 (Background 2 Theme Color)	Specifies that the color to be used shall be the theme color specified by the bg2 attribute of the clrSchemeMapping element (§17.15.1.20).
dark1 (Dark 1 Theme Color)	Specifies that the color to be used shall be the theme color specified by the t1 attribute of the clrSchemeMapping element (§17.15.1.20).
dark2 (Dark 2 Theme Color)	Specifies that the color to be used shall be the theme color specified by the t2 attribute of the clrSchemeMapping element (§17.15.1.20).
followedHyperlink (Followed Hyperlink Theme Color)	Specifies that the color to be used shall be the theme color specified by the followedHyperlink attribute of the clrSchemeMapping element (§17.15.1.20).
hyperlink (Hyperlink Theme Color)	Specifies that the color to be used shall be the theme color specified by the hyperlink attribute of the clrSchemeMapping element (§17.15.1.20).
light1 (Light 1 Theme Color)	Specifies that the color to be used shall be the theme color specified by the bg1 attribute of the clrSchemeMapping element (§17.15.1.20).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
light2 (Light 2 Theme Color)	Specifies that the color to be used shall be the theme color specified by the bg2 attribute of the clrSchemeMapping element (§17.15.1.20).
none (No Theme Color)	Specifies that no theme color shall be applied to the current object.
text1 (Text 1 Theme Color)	Specifies that the color to be used shall be the theme color specified by the t1 attribute of the clrSchemeMapping element (§17.15.1.20).
text2 (Text 2 Theme Color)	Specifies that the color to be used shall be the theme color specified by the t2 attribute of the clrSchemeMapping element (§17.15.1.20).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ThemeColor) is located in §A.1. *end note*]

17.18.98 ST_UcharHexNumber (Two Digit Hexadecimal Value)

This simple type specifies a number value specified as a one octet (two digit) hexadecimal number, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following value for a node of type ST_UcharHexNumber: BE.

This value is permitted, as it contains two hexadecimal digits, as an encoding of an octet of the actual decimal number value. *end example*]

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 2 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UcharHexNumber) is located in §A.1. *end note*]

17.18.99 ST_Underline (Underline Patterns)

This simple type specifies the types of patterns which can be used to create the underline applied beneath the text in a run.

[*Example*: Consider a run of text which must have a double underline explicitly turned on for the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:u w:val="double"/>
</w:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The val of the underline on this run is `double`, so the style of the underline on this run must be a double line. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dash (Dashed Underline)	Specifies an underline consisting of a dashed line beneath all characters in this run. [*Example*: Underline text. *end example*]
dashDotDotHeavy (Thick Dash-Dot-Dot Underline)	Specifies an underline consisting of a series of thick dash, dot, dot characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dashDotHeavy (Thick Dash-Dot Underline)	Specifies an underline consisting of a series of thick dash, dot characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dashedHeavy (Thick Dashed Underline)	Specifies an underline consisting of a series of thick dashes beneath all characters in this run. [*Example*: Underline text. *end example*]
dashLong (Long Dashed Underline)	Specifies an underline consisting of long dashed characters beneath all characters in this run. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Underline text. *end example*]
dashLongHeavy (Thick Long Dashed Underline)	Specifies an underline consisting of thick long dashed characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dotDash (Dash-Dot Underline)	Specifies an underline consisting of a series of dash, dot characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dotDotDash (Dash-Dot-Dot Underline)	Specifies an underline consisting of a series of dash, dot, dot characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dotted (Dotted Underline)	Specifies an underline consisting of a series of dot characters beneath all characters in this run. [*Example*: Underline text. *end example*]
dottedHeavy (Thick Dotted Underline)	Specifies an underline consisting of a series of thick dot characters beneath all characters in this run. [*Example*: Underline text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	end example]
double (Double Underline)	Specifies an underline consisting of two lines beneath all characters in this run. [*Example*: Underline text. *end example*]
none (No Underline)	Specifies no underline beneath this run. [*Example*: Underline text. *end example*]
single (Single Underline)	Specifies an underline consisting of a single line beneath all characters in this run. [*Example*: Underline text. *end example*]
thick (Thick Underline)	Specifies an underline consisting of a single thick line beneath all characters in this run. [*Example*: Underline text. *end example*]
wave (Wave Underline)	Specifies an underline consisting of a single wavy line beneath all characters in this run. [*Example*: Underline text. *end example*]
wavyDouble (Double Wave Underline)	Specifies an underline consisting of a pair of wavy lines beneath all characters in this run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: Underline text. *end example*]
wavyHeavy (Heavy Wave Underline)	Specifies an underline consisting of a single thick wavy line beneath all characters in this run. [*Example*: Underline text. *end example*]
words (Underline Non-Space Characters Only)	Specifies an underline consisting of a single line beneath all non-space characters in the run. There shall be no underline beneath any space character (breaking or non-breaking). [*Example*: Underline text. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Underline) is located in §A.1. *end note*]

17.18.100 ST_VAnchor (Vertical Anchor Location)

This simple type specifies the vertical position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final vertical position of the object in the document.

[*Example*: Consider a text frame which should be positioned one inch to the right of its column in a left-to-right document. This text frame would be specified using the following WordprocessingML:

```
<w:pPr>
  <w:framePr … w:y="1440" w:vAnchor="page" />
</w:pPr>
```

These frame vertical anchor properties specify that they are relative to the anchor paragraph's page. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
margin (Relative To Margin)	Specifies that the parent object shall be vertically anchored to the text margins. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the text margin.
page (Relative To Page)	Specifies that the parent object shall be vertically anchored to the page edge. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the edge of the page.
text (Relative To Vertical Text Extents)	Specifies that the parent object shall be vertically anchored to the text extents. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the top edge of the text in the anchor paragraph.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VAnchor) is located in §A.1. *end note*]

17.18.101 ST_VerticalJc (Vertical Alignment Type)

This simple type specifies the vertical alignment for text between the top and bottom margins of the parent container (page or table cell).

[*Example*: Consider a region where the text must be vertically centered in the parent element. This would require a val value of center, in order to specify that all justification vertically must be centered relative to the parent. For a section, this setting would be specified as follows:

```
<w:vAlign w:val="center" />
```

The val attribute of center specifies that the content is centered relative to its container . *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
both (Vertical Justification)	Specifies that the text shall be vertically justified

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	between the top and bottom margins of the parent object, by adding additional line spacing to each paragraph as required. This setting is only applied for the content of the section which is displayed on full pages. If the content does not use the full page (e.g. another section begins on the same page, or the document ends mid-page), then the value shall be ignored when rendering that page (returning to the default value of top) This value is only permitted for page justification settings, and shall be ignored when specified on a table cell (returning to the default value of top).
bottom (Align Bottom)	Specifies that the text shall be vertically aligned to the bottom margin of the parent object, by moving all text to the bottom text extent within the parent object as required.
center (Align Center)	Specifies that the text shall be vertically aligned to the center of the parent object..
top (Align Top)	Specifies that the text shall be vertically aligned to the top margin of the parent object, by moving all text to the top text extent within the parent object as required.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalJc) is located in §A.1. *end note*]

17.18.102 ST_View (Document View Values)

This simple type defines the kinds of view available to an application when rendering a WordprocessingML document. Those view kinds are, as follows: *default view*, *draft view*, *outline view*, *print layout view*, and *web page view*.

[*Example*: Consider a WordprocessingML document that is to be displayed on a screen in the same form as that document would be printed. This intent is be specified using the following WordprocessingML in the document settings part:

```
<w:view w:val="print" />
```

end example]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
masterPages (Master Document View)	Specifies that a given WordprocessingML document should be rendered in a manner that allows a (possibly) long document to be viewed, navigated, or edited in an outline form, possibly based on the headings in that document, showing the relationships among master documents and subdocuments (see §17.17.1).
none (Default View)	Specifies that a given WordprocessingML document should be rendered in the default view of the application.
normal (Draft View)	Specifies that a given WordprocessingML document should be rendered in a manner that allows a (possibly) long document to be viewed, navigated, or edited.
outline (Outline View)	Specifies that a given WordprocessingML document should be rendered in a manner that allows a (possibly) long document to be viewed, navigated, or edited in an outline form, possibly based on the headings in that document.
print (Print Layout View)	Specifies that a given WordprocessingML document should be rendered in a view mimicking the way that document would be printed.
web (Web Page View)	Specifies that a given WordprocessingML document should be rendered in a view mimicking the way that document would be displayed as a web page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_View) is located in §A.1. *end note*]

17.18.103 ST_WmlColorSchemeIndex (Theme Color Reference)

This simple type specifies the possible set of theme color stored in the document's Theme part which can be referenced by document content. This reference is used to map the use of the theme colors in the ST_ThemeColor enumeration to the theme colors in the theme part.

[*Example*: Consider a WordprocessingML document that must have references to the theme color `accent1` mapped to the theme color lt1 as defined in the document's theme part. This requirement would be specified using the following WordprocessingML in the document settings:

```
<w:clrSchemeMapping w:accent1="light1" />
```


©ISO/IEC 2016 – All rights reserved

The accent1 attribute has a value of `light1`, specifying that uses of the theme color value `accent1` must be mapped to the theme color lt1. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent 1 Theme Color Reference)	Specifies a reference to the accent1 theme color in the document's Theme part.
accent2 (Accent 2 Theme Color Reference)	Specifies a reference to the accent2 theme color in the document's Theme part.
accent3 (Accent 3 Theme Color Reference)	Specifies a reference to the accent3 theme color in the document's Theme part.
accent4 (Accent4 Theme Color Reference)	Specifies a reference to the accent4 theme color in the document's Theme part.
accent5 (Accent5 Theme Color Reference)	Specifies a reference to the accent5 theme color in the document's Theme part.
accent6 (Accent 6 Theme Color Reference)	Specifies a reference to the accent6 theme color in the document's Theme part.
dark1 (Dark 1 Theme Color Reference)	Specifies a reference to the dk1 theme color in the document's Theme part.
dark2 (Dark 2 Theme Color Reference)	Specifies a reference to the dk2 theme color in the document's Theme part.
followedHyperlink (Followed Hyperlink Theme Color Reference)	Specifies a reference to the folHlink theme color in the document's Theme part.
hyperlink (Hyperlink Theme Color Reference)	Specifies a reference to the hlink theme color in the document's Theme part.
light1 (Light 1 Theme Color Reference)	Specifies a reference to the lt1 theme color in the document's Theme part.
light2 (Light 2 Theme Color Reference)	Specifies a reference to the lt2 theme color in the document's Theme part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WmlColorSchemeIndex) is located in §A.1. *end note*]

17.18.104 ST_Wrap (Text Wrapping around Text Frame Type)

This simple type specifies the type of text wrapping which shall be allowed around a text frame within a document.

[*Example*: Consider the following WordprocessingML fragment specifying a text frame:

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:x="1643" w:y="73" />
  </w:pPr>
  <w:r>
    <w:t>Text Frame Content.</w:t>
  </w:r>
</w:p>
```

This wrap attribute on this text frame specifies that when the frame is rendered on the page, any non-text frame paragraphs which would normally flow onto the same lines must be allowed to wrap around it. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
around (Allow Text Wrapping Around Frame)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document.
auto (Default Text Wrapping Around Frame)	Specifies that text shall have the default application-defined behavior of the application displaying the WordprocessingML document with regard to the text wrapping displayed around the frame.
none (No Text Wrapping Around Frame)	Specifies that text shall not be allowed to wrap around the remaining space on each lines around this text frame. Any text content shall therefore be placed on the next line following this text frame which does not intersect with the frame's extents.
notBeside (No Text Wrapping Beside Frame)	Specifies that text shall not be allowed to wrap around the remaining space on each lines around this text frame. Any text content shall therefore be placed on the next line following this text frame which does not intersect with the frame's extents.
through (Through Text Wrapping Around Frame)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document.
tight (Tight Text Wrapping Around Frame)	Specifies that text shall be allowed to tightly wrap around the remaining space on each line around this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	text frame in the document.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Wrap) is located in §A.1. *end note*]

17.18.105 ST_Zoom (Magnification Preset Values)

This simple type specifies the type of magnification settings which can be applied to a given document on open.

[*Example*: Consider a WordprocessingML document that should be visible without any horizontal scrolling when it is displayed. This requirement would be specified using the following WordprocessingML:

```
<w:zoom w:val="bestFit" w:percent="90%" />
```

The val attribute is equal to the value `bestFit` specifying that an application must dynamically calculate the magnification needed such that the given document must be visible on the horizontal plane of the document with no horizontal scrolling required to see any part of the WordprocessingML document's pages. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bestFit (Display Page Width)	Specifies that the magnification setting shall be adjusted to ensure the width of the current page matches the available window width.
fullPage (Display One Full Page)	Specifies that the magnification setting shall be adjusted to ensure that one full page can be seen at a time.
none (No Preset Magnification)	Specifies that no preset magnification is present, and the last known cached setting shall be used.
textFit (Display Text Width)	Specifies that the magnification setting shall be adjusted to ensure the width of the text extents on the current page matches the available window width.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Zoom) is located in §A.1. *end note*]

17.18.106 ST_TextScalePercent (Text Expansion/Compression Percentage)

This simple type specifies that the percentage by which the contents of a run shall be expanded or compressed with respect to its normal (100%) character width, with a minimum width of 1% and maximum width of 600%.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a run of text that must be compressed by half when displaying each character within the contents of the run. This constraint is specified using the following WordprocessingML:

```
<w:rPr>
  <w:w w:val="50%"/>
</w:rPr>
```

This run explicitly declares that the w value is 50%, so the contents of this run appear at 50% of their normal character width by compressing the width of each character. *end example*]

This simple type also specifies the following restrictions:

This simple type's contents shall match the following regular expression pattern:
0*(600|([0-5]?[0-9]?[0-9]))%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextScalePercent) is located in §A.1. *end note*]

17.18.107 ST_MeasurementOrPercent (Measurement or Percentage Value)

This simple type specifies the possible values for a table measurement, which can be percentage-based or absolute. See the union's member types for details.

This simple type is a union of the following types:

The ST_DecimalNumberOrPercent simple type (§17.18.11).
The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MeasurementOrPercent) is located in §A.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18. SpreadsheetML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, “Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)”. *end note*]

The subordinate subclauses specify the semantics for the XML markup comprising a SpreadsheetML document, as defined by §12 of this Part of ISO/IEC 29500.

18.1 Table of Contents

This subclause is informative.

18.2 Workbook 1542
18.2.1 bookViews (Workbook Views) 1543
18.2.2 calcPr (Calculation Properties) 1543
18.2.3 customWorkbookView (Custom Workbook View) 1547
18.2.4 customWorkbookViews (Custom Workbook Views) 1553
18.2.5 definedName (Defined Name) 1553
18.2.6 definedNames (Defined Names) 1558
18.2.7 ext (Extension) 1558
18.2.8 externalReference (External Reference) 1560
18.2.9 externalReferences (External References) 1560
18.2.10 extLst (Future Feature Data Storage Area) 1560
18.2.11 fileRecoveryPr (File Recovery Properties) 1561
18.2.12 fileSharing (File Sharing) 1562
18.2.13 fileVersion (File Version) 1565
18.2.14 functionGroup (Function Group) 1566
18.2.15 functionGroups (Function Groups) 1566
18.2.16 oleSize (Embedded Object Size) 1567
18.2.17 pivotCache (PivotCache) 1567
18.2.18 pivotCaches (PivotCaches) 1568
18.2.19 sheet (Sheet Information) 1568
18.2.20 sheets (Sheets) 1569
18.2.21 smartTagPr (Smart Tag Properties) 1569
18.2.22 smartTagType (Smart Tag Type) 1570
18.2.23 smartTagTypes (Smart Tag Types) 1571
18.2.24 webPublishing (Web Publishing Properties) 1571
18.2.25 webPublishObject (Web Publishing Object) 1572
18.2.26 webPublishObjects (Web Publish Objects) 1573
18.2.27 workbook (Workbook) 1574
18.2.28 workbookPr (Workbook Properties) 1576
18.2.29 workbookProtection (Workbook Protection) 1580
18.2.30 workbookView (Workbook View) 1587

18.3 Worksheets 1589

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.3.1 Worksheets 1590
18.3.1.1 anchor (Object Cell Anchor) 1590
18.3.1.2 autoFilter (AutoFilter Settings) 1591
18.3.1.3 brk (Break) 1592
18.3.1.4 c (Cell) 1593
18.3.1.5 cellSmartTag (Cell Smart Tag) 1594
18.3.1.6 cellSmartTagPr (Smart Tag Properties) 1595
18.3.1.7 cellSmartTags (Cell Smart Tags) 1595
18.3.1.8 cellWatch (Cell Watch Item) 1596
18.3.1.9 cellWatches (Cell Watch Items) 1596
18.3.1.10 cfRule (Conditional Formatting Rule) 1597
18.3.1.11 cfvo (Conditional Format Value Object) 1599
18.3.1.12 chartsheet (Chart Sheet) 1600
18.3.1.13 col (Column Width & Formatting) 1600
18.3.1.14 colBreaks (Vertical Page Breaks) 1602
18.3.1.15 color (Data Bar Color) 1603
18.3.1.16 colorScale (Color Scale) 1604
18.3.1.17 cols (Column Information) 1605
18.3.1.18 conditionalFormatting (Conditional Formatting) 1605
18.3.1.19 control (Embedded Control) 1606
18.3.1.20 controlPr (Embedded Control Properties) 1606
18.3.1.21 controls (Embedded Controls) 1611
18.3.1.22 customPr (Custom Property) 1611
18.3.1.23 customProperties (Custom Properties) 1612
18.3.1.24 customSheetView (Custom Chart Sheet View) 1612
18.3.1.25 customSheetView (Custom Sheet View) 1613
18.3.1.26 customSheetViews (Custom Chart Sheet Views) 1615
18.3.1.27 customSheetViews (Custom Sheet Views) 1616
18.3.1.28 dataBar (Data Bar) 1616
18.3.1.29 dataConsolidate (Data Consolidate) 1617
18.3.1.30 dataRef (Data Consolidation Reference) 1618
18.3.1.31 dataRefs (Data Consolidation References) 1618
18.3.1.32 dataValidation (Data Validation) 1619
18.3.1.33 dataValidations (Data Validations) 1620
18.3.1.34 dialogsheet (Dialog Sheet) 1622
18.3.1.35 dimension (Worksheet Dimensions) 1622
18.3.1.36 drawing (Drawing) 1622
18.3.1.37 drawingHF (Drawing Reference in Header Footer) 1623
18.3.1.38 evenFooter (Even Page Footer) 1629
18.3.1.39 evenHeader (Even Page Header) 1629
18.3.1.40 f (Formula) 1630
18.3.1.41 firstFooter (First Page Footer) 1633
18.3.1.42 firstHeader (First Page Header) 1634
18.3.1.43 formula (Formula) 1635
18.3.1.44 formula1 (Formula 1) 1635
18.3.1.45 formula2 (Formula 2) 1636
18.3.1.46 headerFooter (Header Footer Settings) 1636
18.3.1.47 hyperlink (Hyperlink) 1641

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.3.1.48 hyperlinks (Hyperlinks) 1642
18.3.1.49 iconSet (Icon Set) 1642
18.3.1.50 ignoredError (Ignored Error) 1643
18.3.1.51 ignoredErrors (Ignored Errors) 1645
18.3.1.52 inputCells (Input Cells) 1645
18.3.1.53 is (Rich Text Inline) 1646
18.3.1.54 mergeCell (Merged Cell) 1646
18.3.1.55 mergeCells (Merge Cells) 1646
18.3.1.56 objectPr (Embedded Object Properties) 1647
18.3.1.57 oddFooter (Odd Page Footer) 1650
18.3.1.58 oddHeader (Odd Header) 1651
18.3.1.59 oleObject (Embedded Object) 1652
18.3.1.60 oleObjects (Embedded Objects) 1653
18.3.1.61 outlinePr (Outline Properties) 1653
18.3.1.62 pageMargins (Page Margins) 1655
18.3.1.63 pageSetup (Page Setup Settings) 1655
18.3.1.64 pageSetup (Chart Sheet Page Setup) 1661
18.3.1.65 pageSetUpPr (Page Setup Properties) 1664
18.3.1.66 pane (View Pane) 1665
18.3.1.67 picture (Background Image) 1665
18.3.1.68 pivotArea (Pivot Area) 1666
18.3.1.69 pivotSelection (PivotTable Selection) 1667
18.3.1.70 printOptions (Print Options) 1671
18.3.1.71 protectedRange (Protected Range) 1671
18.3.1.72 protectedRanges (Protected Ranges) 1675
18.3.1.73 row (Row) 1675
18.3.1.74 rowBreaks (Horizontal Page Breaks (Row)) 1681
18.3.1.75 scenario (Scenario) 1681
18.3.1.76 scenarios (Scenarios) 1682
18.3.1.77 securityDescriptor (Security Descriptor) 1683
18.3.1.78 selection (Selection) 1684
18.3.1.79 sheetCalcPr (Sheet Calculation Properties) 1684
18.3.1.80 sheetData (Sheet Data) 1685
18.3.1.81 sheetFormatPr (Sheet Format Properties) 1685
18.3.1.82 sheetPr (Sheet Properties) 1686
18.3.1.83 sheetPr (Chart Sheet Properties) 1688
18.3.1.84 sheetProtection (Chart Sheet Protection) 1688
18.3.1.85 sheetProtection (Sheet Protection Options) 1691
18.3.1.86 sheetView (Chart Sheet View) 1697
18.3.1.87 sheetView (Worksheet View) 1698
18.3.1.88 sheetViews (Sheet Views) 1702
18.3.1.89 sheetViews (Chart Sheet Views) 1702
18.3.1.90 smartTags (Smart Tags) 1702
18.3.1.91 sortCondition (Sort Condition) 1703
18.3.1.92 sortState (Sort State) 1704
18.3.1.93 tabColor (Sheet Tab Color) 1705
18.3.1.94 tablePart (Table Part) 1706
18.3.1.95 tableParts (Table Parts) 1707

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.3.1.96 v (Cell Value) 1707
18.3.1.97 webPublishItem (Web Publishing Item) 1708
18.3.1.98 webPublishItems (Web Publishing Items) 1709
18.3.1.99 worksheet (Worksheet) 1710
18.3.2 AutoFilter Settings 1710
18.3.2.1 colorFilter (Color Filter Criteria) 1710
18.3.2.2 customFilter (Custom Filter Criteria) 1711
18.3.2.3 customFilters (Custom Filters) 1711
18.3.2.4 dateGroupItem (Date Grouping) 1712
18.3.2.5 dynamicFilter (Dynamic Filter) 1713
18.3.2.6 filter (Filter) 1714
18.3.2.7 filterColumn (AutoFilter Column) 1715
18.3.2.8 filters (Filter Criteria) 1715
18.3.2.9 iconFilter (Icon Filter) 1716
18.3.2.10 top10 (Top 10) 1716

18.4 Shared String Table 1717

18.4.1 charset (Character Set) 1719
18.4.2 outline (Outline) 1720
18.4.3 phoneticPr (Phonetic Properties) 1721
18.4.4 r (Rich Text Run) 1722
18.4.5 rFont (Font) 1722
18.4.6 rPh (Phonetic Run) 1723
18.4.7 rPr (Run Properties) 1723
18.4.8 si (String Item) 1723
18.4.9 sst (Shared String Table) 1724
18.4.10 strike (Strike Through) 1724
18.4.11 sz (Font Size) 1725
18.4.12 t (Text) 1725
18.4.13 u (Underline) 1726
18.4.14 vertAlign (Vertical Alignment) 1726

18.5 Tables 1726

18.5.1 Tables 1727
18.5.1.1 calculatedColumnFormula (Calculated Column Formula) 1728
18.5.1.2 table (Table) 1728
18.5.1.3 tableColumn (Table Column) 1733
18.5.1.4 tableColumns (Table Columns) 1735
18.5.1.5 tableStyleInfo (Table Style) 1736
18.5.1.6 totalsRowFormula (Totals Row Formula) 1737
18.5.1.7 xmlColumnPr (XML Column Properties) 1737
18.5.2 Single Cell Tables 1739
18.5.2.1 singleXmlCell (Table Properties) 1739
18.5.2.2 singleXmlCells (Single Cells) 1740
18.5.2.3 xmlCellPr (Cell Properties) 1740
18.5.2.4 xmlPr (Column XML Properties) 1741

18.6 Calculation Chain 1742

18.6.1 c (Cell) 1744

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.6.2 calcChain (Calculation Chain Info) 1745

18.7 Comments 1745

18.7.1 author (Author) 1746
18.7.2 authors (Authors) 1747
18.7.3 comment (Comment) 1747
18.7.4 commentList (List of Comments) 1748
18.7.5 commentPr (Comment Properties) 1748
18.7.6 comments (Comments) 1752
18.7.7 text (Comment Text) 1752

18.8 Styles 1752

18.8.1 alignment (Alignment) 1752
18.8.2 b (Bold) 1755
18.8.3 bgColor (Background Color) 1756
18.8.4 border (Border) 1757
18.8.5 borders (Borders) 1758
18.8.6 bottom (Bottom Border) 1759
18.8.7 cellStyle (Cell Style) 1759
18.8.8 cellStyles (Cell Styles) 1761
18.8.9 cellStyleXfs (Formatting Records) 1761
18.8.10 cellXfs (Cell Formats) 1762
18.8.11 colors (Colors) 1763
18.8.12 condense (Condense) 1763
18.8.13 diagonal (Diagonal) 1763
18.8.14 dxf (Formatting) 1763
18.8.15 dxfs (Formats) 1764
18.8.16 end (Trailing Edge Border) 1764
18.8.17 extend (Extend) 1764
18.8.18 family (Font Family) 1765
18.8.19 fgColor (Foreground Color) 1765
18.8.20 fill (Fill) 1766
18.8.21 fills (Fills) 1767
18.8.22 font (Font) 1767
18.8.23 fonts (Fonts) 1768
18.8.24 gradientFill (Gradient) 1768
18.8.25 horizontal (Horizontal Inner Borders) 1771
18.8.26 i (Italic) 1771
18.8.27 indexedColors (Color Indexes) 1772
18.8.28 mruColors (MRU Colors) 1775
18.8.29 name (Font Name) 1775
18.8.30 numFmt (Number Format) 1776
18.8.31 numFmts (Number Formats) 1783
18.8.32 patternFill (Pattern) 1792
18.8.33 protection (Protection Properties) 1792
18.8.34 rgbColor (RGB Color) 1793
18.8.35 scheme (Scheme) 1793
18.8.36 shadow (Shadow) 1793
18.8.37 start (Leading Edge Border) 1794

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.8.38 stop (Gradient Stop) 1794
18.8.39 styleSheet (Style Sheet) 1794
18.8.40 tableStyle (Table Style) 1795
18.8.41 tableStyleElement (Table Style) 1796
18.8.42 tableStyles (Table Styles) 1798
18.8.43 top (Top Border) 1798
18.8.44 vertical (Vertical Inner Border) 1799
18.8.45 xf (Format) 1799

18.9 Metadata 1801

18.9.1 bk (Metadata Block) 1802
18.9.2 bk (Future Metadata Block) 1802
18.9.3 cellMetadata (Cell Metadata) 1803
18.9.4 futureMetadata (Future Metadata) 1803
18.9.5 k (KPI MDX Metadata) 1804
18.9.6 mdx (MDX Metadata Record) 1804
18.9.7 mdxMetadata (MDX Metadata Information) 1805
18.9.8 metadata (Metadata) 1805
18.9.9 metadataStrings (Metadata String Store) 1805
18.9.10 metadataType (Metadata Type Information) 1805
18.9.11 metadataTypes (Metadata Types Collection) 1811
18.9.12 ms (Set MDX Metadata) 1812
18.9.13 n (Member Unique Name Index) 1812
18.9.14 p (Member Property MDX Metadata) 1812
18.9.15 rc (Metadata Record) 1813
18.9.16 t (Tuple MDX Metadata) 1813
18.9.17 valueMetadata (Value Metadata) 1815

18.10 Pivot Tables 1815

18.10.1 Pivot Tables 1820
18.10.1.1 autoSortScope (AutoSort Scope) 1820
18.10.1.2 b (Boolean) 1820
18.10.1.3 cacheField (PivotCache Field) 1821
18.10.1.4 cacheFields (PivotCache Fields) 1824
18.10.1.5 cacheHierarchies (PivotCache Hierarchies) 1825
18.10.1.6 cacheHierarchy (PivotCache Hierarchy) 1825
18.10.1.7 cacheSource (PivotCache Source Description) 1830
18.10.1.8 calculatedItem (Calculated Item) 1831
18.10.1.9 calculatedItems (Calculated Items) 1832
18.10.1.10 calculatedMember (Calculated Member) 1832
18.10.1.11 calculatedMembers (Calculated Members) 1833
18.10.1.12 chartFormat (PivotChart Format) 1834
18.10.1.13 chartFormats (PivotChart Formats) 1835
18.10.1.14 colFields (Column Fields) 1837
18.10.1.15 colHierarchiesUsage (Column OLAP Hierarchy References) 1837
18.10.1.16 colHierarchyUsage (Column OLAP Hierarchies) 1838
18.10.1.17 colItems (Column Items) 1838
18.10.1.18 conditionalFormat (Conditional Formatting) 1840
18.10.1.19 conditionalFormats (Conditional Formats) 1840

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.20 consolidation (Consolidation Source) 1841
18.10.1.21 d (Date Time) 1842
18.10.1.22 dataField (Data Field Item) 1842
18.10.1.23 dataFields (Data Fields) 1844
18.10.1.24 dimension (OLAP Dimension) 1845
18.10.1.25 dimensions (OLAP Dimensions) 1845
18.10.1.26 discretePr (Discrete Grouping Properties) 1846
18.10.1.27 e (Error Value) 1848
18.10.1.28 entries (Entries) 1850
18.10.1.29 field (Field) 1851
18.10.1.30 fieldGroup (Field Group Properties) 1851
18.10.1.31 fieldsUsage (Fields Usage) 1853
18.10.1.32 fieldUsage (PivotCache Field Id) 1853
18.10.1.33 filter (PivotTable Advanced Filter) 1854
18.10.1.34 filters (Filters) 1855
18.10.1.35 format (PivotTable Format) 1856
18.10.1.36 formats (PivotTable Formats) 1857
18.10.1.37 group (OLAP Group) 1857
18.10.1.38 groupItems (OLAP Group Items) 1858
18.10.1.39 groupLevel (OLAP Grouping Levels) 1859
18.10.1.40 groupLevels (OLAP Grouping Levels) 1861
18.10.1.41 groupMember (OLAP Group Member) 1862
18.10.1.42 groupMembers (OLAP Group Members) 1862
18.10.1.43 groups (OLAP Level Groups) 1863
18.10.1.44 i (Row Items) 1863
18.10.1.45 item (PivotTable Field Item) 1864
18.10.1.46 items (Field Items) 1867
18.10.1.47 kpi (OLAP KPI) 1868
18.10.1.48 kpis (OLAP KPIs) 1869
18.10.1.49 location (PivotTable Location) 1870
18.10.1.50 m (No Value) 1871
18.10.1.51 map (OLAP Measure Group) 1873
18.10.1.52 maps (OLAP Measure Group) 1874
18.10.1.53 measureGroup (OLAP Measure Group) 1874
18.10.1.54 measureGroups (OLAP Measure Groups) 1875
18.10.1.55 member (Member) 1876
18.10.1.56 members (Members) 1876
18.10.1.57 mp (OLAP Member Property) 1876
18.10.1.58 mpMap (Member Properties Map) 1878
18.10.1.59 mps (OLAP Member Properties) 1879
18.10.1.60 n (Numeric) 1879
18.10.1.61 page (Page Items) 1881
18.10.1.62 pageField (Page Field) 1882
18.10.1.63 pageFields (Page Field Items) 1883
18.10.1.64 pageItem (Page Item) 1883
18.10.1.65 pages (Page Item Values) 1883
18.10.1.66 pivotAreas (Pivot Areas) 1884
18.10.1.67 pivotCacheDefinition (PivotCache Definition) 1884

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.10.1.68 pivotCacheRecords (PivotCache Records) 1888
18.10.1.69 pivotField (PivotTable Field) 1889
18.10.1.70 pivotFields (PivotTable Fields) 1900
18.10.1.71 pivotHierarchies (PivotTable OLAP Hierarchies) 1901
18.10.1.72 pivotHierarchy (OLAP Hierarchy) 1902
18.10.1.73 pivotTableDefinition (PivotTable Definition) 1904
18.10.1.74 pivotTableStyleInfo (PivotTable Style) 1934
18.10.1.75 query (Query) 1936
18.10.1.76 queryCache (OLAP Query Cache) 1936
18.10.1.77 r (PivotCache Record) 1936
18.10.1.78 rangePr (Range Grouping Properties) 1937
18.10.1.79 rangeSet (Range Set) 1938
18.10.1.80 rangeSets (Range Sets) 1941
18.10.1.81 rowFields (Row Fields) 1942
18.10.1.82 rowHierarchiesUsage (Row OLAP Hierarchy References) 1943
18.10.1.83 rowHierarchyUsage (Row OLAP Hierarchies) 1943
18.10.1.84 rowItems (Row Items) 1943
18.10.1.85 s (Character Value) 1945
18.10.1.86 serverFormat (Server Format) 1947
18.10.1.87 serverFormats (Server Formats) 1947
18.10.1.88 set (OLAP Set) 1947
18.10.1.89 sets (Sets) 1948
18.10.1.90 sharedItems (Shared Items) 1949
18.10.1.91 sortByTuple (Sort By Tuple) 1952
18.10.1.92 tpl (Tuple) 1953
18.10.1.93 tpls (Tuples) 1953
18.10.1.94 tupleCache (Tuple Cache) 1953
18.10.1.95 worksheetSource (Worksheet PivotCache Source) 1953
18.10.1.96 x (Member Property Index) 1954
18.10.1.97 x (Shared Items Index) 1954
18.10.2 Shared Pivot Table Data 1955
18.10.2.1 reference (Reference) 1955
18.10.2.2 references (References) 1959

18.11 Shared Workbook Data 1959

18.11.1 Shared Workbook Data 1960
18.11.1.1 header (Header) 1962
18.11.1.2 headers (Revision Headers) 1964
18.11.1.3 nc (New Cell Data) 1966
18.11.1.4 ndxf (New Formatting Information) 1967
18.11.1.5 oc (Old Cell Data) 1967
18.11.1.6 odxf (Old Formatting Information) 1968
18.11.1.7 oldFormula (Old Formula) 1968
18.11.1.8 raf (Revision AutoFormat) 1968
18.11.1.9 rcc (Revision Cell Change) 1970
18.11.1.10 rcft (Revision Merge Conflict) 1971
18.11.1.11 rcmt (Revision Cell Comment) 1972
18.11.1.12 rcv (Revision Custom View) 1973

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.11.1.13 rdn (Revision Defined Name) 1974
18.11.1.14 reviewed (Reviewed) 1977
18.11.1.15 reviewedList (Reviewed List) 1977
18.11.1.16 revisions (Revisions) 1978
18.11.1.17 rfmt (Revision Format) 1979
18.11.1.18 ris (Revision Insert Sheet) 1980
18.11.1.19 rm (Revision Cell Move) 1981
18.11.1.20 rqt (Revision Query Table) 1982
18.11.1.21 rrc (Revision Row Column Insert Delete) 1982
18.11.1.22 rsnm (Revision Sheet Name) 1984
18.11.1.23 sheetId (Sheet Id) 1985
18.11.1.24 sheetIdMap (Sheet Id Map) 1985
18.11.1.25 undo (Undo) 1985
18.11.2 Shared Workbook User Data 1987
18.11.2.1 userInfo (User Information) 1987
18.11.2.2 users (User List) 1987

18.12 QueryTable Data 1988

18.12.1 deletedField (Deleted Field) 1989
18.12.2 queryTable (Query Table) 1989
18.12.3 queryTableDeletedFields (Deleted Fields) 1992
18.12.4 queryTableField (QueryTable Field) 1993
18.12.5 queryTableFields (Query table fields) 1994
18.12.6 queryTableRefresh (QueryTable Refresh Information) 1994

18.13 External Data Connections 1995

18.13.1 connection (Connection) 1996
18.13.2 connections (Connections) 2000
18.13.3 dbPr (Database Properties) 2000
18.13.4 m (No Value) 2003
18.13.5 olapPr (OLAP Properties) 2003
18.13.6 parameter (Parameter Properties) 2006
18.13.7 parameters (Query Parameters) 2008
18.13.8 s (Character Value) 2009
18.13.9 tables (Tables) 2009
18.13.10 textField (Text Import Field Settings) 2009
18.13.11 textFields (Fields) 2010
18.13.12 textPr (Text Import Settings) 2010
18.13.13 webPr (Web Query Properties) 2013

18.14 Supplementary Workbook Data 2016

18.14.1 cell (External Cell Data) 2017
18.14.2 ddeItem (DDE Item definition) 2018
18.14.3 ddeItems (DDE Items Collection) 2019
18.14.4 ddeLink (DDE Connection) 2019
18.14.5 definedName (Defined Name) 2020
18.14.6 definedNames (Named Links) 2020
18.14.7 externalBook (External Workbook) 2020
18.14.8 externalLink (External Reference) 2021

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.14.9 oleItem (Object Link Item) 2021
18.14.10 oleItems (Object Link Items) 2022
18.14.11 oleLink (Generic Object Link Connection) 2022
18.14.12 row (Row) 2023
18.14.13 sheetData (External Sheet Data Set) 2023
18.14.14 sheetDataSet (Cached Worksheet Data) 2024
18.14.15 sheetName (Sheet Name) 2024
18.14.16 sheetNames (Supporting Workbook Sheet Names) 2024
18.14.17 val (DDE Link Value) 2024
18.14.18 value (Value) 2025
18.14.19 values (DDE Name Values) 2026

18.15 Volatile Dependencies 2026

18.15.1 main (Main) 2028
18.15.2 stp (Strings in Subtopic) 2029
18.15.3 tp (Topic) 2029
18.15.4 tr (References) 2030
18.15.5 volType (Volatile Dependency Type) 2030
18.15.6 volTypes (Volatile Dependency Types) 2031

18.16 Custom XML Mappings 2031

18.16.1 DataBinding (XML Mapping) 2034
18.16.2 Map (XML Mapping Properties) 2036
18.16.3 MapInfo (XML Mapping) 2037
18.16.4 Schema (XML Schema) 2038

18.17 Formulas 2039

18.17.1 Introduction 2039
18.17.2 Syntax 2040
18.17.2.1 Constants 2045
18.17.2.2 Operators 2047
18.17.2.3 Cell References 2049
18.17.2.4 Functions 2055
18.17.2.5 Names 2057
18.17.2.6 Types and Values 2058
18.17.2.7 Single- and Array Formulas 2059
18.17.3 Error values 2060
18.17.4 Dates and Times 2062
18.17.4.1 Date Conversion for Serial Date-Times 2063
18.17.4.2 Time Conversion for Serial Date-Times 2064
18.17.4.3 Combined Date and Time Conversion for Serial Date-Times 2064
18.17.5 Limits and Precision 2065
18.17.5.1 Limits 2065
18.17.5.2 Precision 2065
18.17.5.3 Lexical Representation 2066
18.17.5.4 Interpretation 2066
18.17.6 XML Representation 2067
18.17.6.1 Cell Reference Style 2067
18.17.6.2 Scalar Formulas 2067

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.6.3 Array Formulas 2068
18.17.6.4 Formula Evaluation Order 2069
18.17.6.5 Name Representation 2070
18.17.6.6 Value Representation 2070
18.17.6.7 Dates and Times 2070
18.17.7 Predefined Function Definitions 2071
18.17.7.1 ABS 2074
18.17.7.2 ACCRINT 2075
18.17.7.3 ACCRINTM 2077
18.17.7.4 ACOS 2080
18.17.7.5 ACOSH 2080
18.17.7.6 ADDRESS 2081
18.17.7.7 AMORDEGRC 2082
18.17.7.8 AMORLINC 2085
18.17.7.9 AND 2088
18.17.7.10 AREAS 2088
18.17.7.11 ASC 2089
18.17.7.12 ASIN 2089
18.17.7.13 ASINH 2090
18.17.7.14 ATAN 2091
18.17.7.15 ATAN2 2091
18.17.7.16 ATANH 2092
18.17.7.17 AVEDEV 2092
18.17.7.18 AVERAGE 2093
18.17.7.19 AVERAGEA 2094
18.17.7.20 AVERAGEIF 2095
18.17.7.21 AVERAGEIFS 2096
18.17.7.22 BAHTTEXT 2097
18.17.7.23 BESSELI 2098
18.17.7.24 BESSELJ 2099
18.17.7.25 BESSELK 2100
18.17.7.26 BESSELY 2100
18.17.7.27 BETADIST 2101
18.17.7.28 BETAINV 2102
18.17.7.29 BIN2DEC 2103
18.17.7.30 BIN2HEX 2104
18.17.7.31 BIN2OCT 2105
18.17.7.32 BINOMDIST 2106
18.17.7.33 CEILING 2107
18.17.7.34 CELL 2108
18.17.7.35 CHAR 2115
18.17.7.36 CHIDIST 2115
18.17.7.37 CHIINV 2116
18.17.7.38 CHITEST 2117
18.17.7.39 CHOOSE 2118
18.17.7.40 CLEAN 2119
18.17.7.41 CODE 2119
18.17.7.42 COLUMN 2120

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.43 COLUMNS 2121
18.17.7.44 COMBIN 2121
18.17.7.45 COMPLEX 2122
18.17.7.46 CONCATENATE 2123
18.17.7.47 CONFIDENCE 2123
18.17.7.48 CONVERT 2124
18.17.7.49 CORREL 2129
18.17.7.50 COS 2130
18.17.7.51 COSH 2131
18.17.7.52 COUNT 2131
18.17.7.53 COUNTA 2132
18.17.7.54 COUNTBLANK 2133
18.17.7.55 COUNTIF 2133
18.17.7.56 COUNTIFS 2134
18.17.7.57 COUPDAYBS 2136
18.17.7.58 COUPDAYS 2138
18.17.7.59 COUPDAYSNC 2141
18.17.7.60 COUPNCD 2143
18.17.7.61 COUPNUM 2146
18.17.7.62 COUPPCD 2148
18.17.7.63 COVAR 2151
18.17.7.64 CRITBINOM 2152
18.17.7.65 CUBEKPIMEMBER 2153
18.17.7.66 CUBEMEMBER 2154
18.17.7.67 CUBEMEMBERPROPERTY 2155
18.17.7.68 CUBERANKEDMEMBER 2156
18.17.7.69 CUBESET 2157
18.17.7.70 CUBESETCOUNT 2159
18.17.7.71 CUBEVALUE 2160
18.17.7.72 CUMIPMT 2161
18.17.7.73 CUMPRINC 2162
18.17.7.74 DATE 2163
18.17.7.75 DATEDIF 2164
18.17.7.76 DATEVALUE 2166
18.17.7.77 DAVERAGE 2167
18.17.7.78 DAY 2169
18.17.7.79 DAYS360 2170
18.17.7.80 DB 2171
18.17.7.81 DCOUNT 2173
18.17.7.82 DCOUNTA 2173
18.17.7.83 DDB 2174
18.17.7.84 DEC2BIN 2175
18.17.7.85 DEC2HEX 2176
18.17.7.86 DEC2OCT 2177
18.17.7.87 DEGREES 2178
18.17.7.88 DELTA 2178
18.17.7.89 DEVSQ 2179
18.17.7.90 DGET 2180

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.91 DISC ... 2180
18.17.7.92 DMAX ... 2183
18.17.7.93 DMIN ... 2184
18.17.7.94 DOLLAR ... 2184
18.17.7.95 DOLLARDE ... 2186
18.17.7.96 DOLLARFR ... 2186
18.17.7.97 DPRODUCT ... 2187
18.17.7.98 DSTDEV ... 2188
18.17.7.99 DSTDEVP ... 2188
18.17.7.100 DSUM ... 2189
18.17.7.101 DURATION ... 2190
18.17.7.102 DVAR ... 2192
18.17.7.103 DVARP ... 2193
18.17.7.104 ECMA.CEILING ... 2194
18.17.7.105 EDATE ... 2194
18.17.7.106 EFFECT ... 2195
18.17.7.107 EOMONTH ... 2196
18.17.7.108 ERF ... 2197
18.17.7.109 ERFC ... 2198
18.17.7.110 ERROR.TYPE ... 2199
18.17.7.111 EVEN ... 2200
18.17.7.112 EXACT ... 2201
18.17.7.113 EXP ... 2201
18.17.7.114 EXPONDIST ... 2202
18.17.7.115 FACT ... 2203
18.17.7.116 FACTDOUBLE ... 2203
18.17.7.117 FALSE ... 2204
18.17.7.118 FDIST ... 2205
18.17.7.119 FIND ... 2206
18.17.7.120 FINDB ... 2206
18.17.7.121 FINV ... 2207
18.17.7.122 FISHER ... 2208
18.17.7.123 FISHERINV ... 2209
18.17.7.124 FIXED ... 2209
18.17.7.125 FLOOR ... 2210
18.17.7.126 FORECAST ... 2211
18.17.7.127 FREQUENCY ... 2212
18.17.7.128 FTEST ... 2213
18.17.7.129 FV ... 2213
18.17.7.130 FVSCHEDULE ... 2214
18.17.7.131 GAMMADIST ... 2215
18.17.7.132 GAMMAINV ... 2216
18.17.7.133 GAMMALN ... 2217
18.17.7.134 GCD ... 2218
18.17.7.135 GEOMEAN ... 2218
18.17.7.136 GESTEP ... 2219
18.17.7.137 GETPIVOTDATA ... 2220
18.17.7.138 GROWTH ... 2221

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.139 HARMEAN 2223
18.17.7.140 HEX2BIN 2224
18.17.7.141 HEX2DEC 2225
18.17.7.142 HEX2OCT 2225
18.17.7.143 HLOOKUP 2226
18.17.7.144 HOUR 2228
18.17.7.145 HYPERLINK 2229
18.17.7.146 HYPGEOMDIST 2230
18.17.7.147 IF 2231
18.17.7.148 IFERROR 2232
18.17.7.149 IMABS 2233
18.17.7.150 IMAGINARY 2233
18.17.7.151 IMARGUMENT 2234
18.17.7.152 IMCONJUGATE 2235
18.17.7.153 IMCOS 2235
18.17.7.154 IMDIV 2236
18.17.7.155 IMEXP 2237
18.17.7.156 IMLN 2238
18.17.7.157 IMLOG10 2238
18.17.7.158 IMLOG2 2239
18.17.7.159 IMPOWER 2240
18.17.7.160 IMPRODUCT 2241
18.17.7.161 IMREAL 2242
18.17.7.162 IMSIN 2242
18.17.7.163 IMSQRT 2243
18.17.7.164 IMSUB 2244
18.17.7.165 IMSUM 2245
18.17.7.166 INDEX 2245
18.17.7.167 INDIRECT 2248
18.17.7.168 INFO 2249
18.17.7.169 INT 2250
18.17.7.170 INTERCEPT 2251
18.17.7.171 INTRATE 2252
18.17.7.172 IPMT 2255
18.17.7.173 IRR 2256
18.17.7.174 ISBLANK 2256
18.17.7.175 ISERR 2257
18.17.7.176 ISERROR 2257
18.17.7.177 ISEVEN 2258
18.17.7.178 ISLOGICAL 2258
18.17.7.179 ISNA 2259
18.17.7.180 ISNONTEXT 2259
18.17.7.181 ISNUMBER 2260
18.17.7.182 ISO.CEILING 2261
18.17.7.183 ISODD 2261
18.17.7.184 ISPMT 2262
18.17.7.185 ISREF 2263
18.17.7.186 ISTEXT 2263

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.187 JIS 2264
18.17.7.188 KURT 2264
18.17.7.189 LARGE 2265
18.17.7.190 LCM 2266
18.17.7.191 LEFT 2266
18.17.7.192 LEFTB 2267
18.17.7.193 LEN 2268
18.17.7.194 LENB 2268
18.17.7.195 LINEST 2269
18.17.7.196 LN 2271
18.17.7.197 LOG 2272
18.17.7.198 LOG10 2272
18.17.7.199 LOGEST 2273
18.17.7.200 LOGINV 2275
18.17.7.201 LOGNORMDIST 2276
18.17.7.202 LOOKUP 2276
18.17.7.203 LOWER 2278
18.17.7.204 MATCH 2279
18.17.7.205 MAX 2280
18.17.7.206 MAXA 2281
18.17.7.207 MDETERM 2281
18.17.7.208 MDURATION 2282
18.17.7.209 MEDIAN 2285
18.17.7.210 MID 2286
18.17.7.211 MIDB 2287
18.17.7.212 MIN 2287
18.17.7.213 MINA 2288
18.17.7.214 MINUTE 2289
18.17.7.215 MINVERSE 2290
18.17.7.216 MIRR 2290
18.17.7.217 MMULT 2291
18.17.7.218 MOD 2292
18.17.7.219 MODE 2293
18.17.7.220 MONTH 2294
18.17.7.221 MROUND 2294
18.17.7.222 MULTINOMIAL 2295
18.17.7.223 N 2296
18.17.7.224 NA 2297
18.17.7.225 NEGBINOMDIST 2297
18.17.7.226 NETWORKDAYS 2298
18.17.7.227 NETWORKDAYS.INTL 2299
18.17.7.228 NOMINAL 2301
18.17.7.229 NORMDIST 2302
18.17.7.230 NORMINV 2303
18.17.7.231 NORMSDIST 2303
18.17.7.232 NORMSINV 2304
18.17.7.233 NOT 2305
18.17.7.234 NOW 2305

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.235 NPER 2305
18.17.7.236 NPV 2306
18.17.7.237 OCT2BIN 2308
18.17.7.238 OCT2DEC 2309
18.17.7.239 OCT2HEX 2309
18.17.7.240 ODD 2310
18.17.7.241 ODDFPRICE 2311
18.17.7.242 ODDFYIELD 2315
18.17.7.243 ODDLPRICE 2318
18.17.7.244 ODDLYIELD 2321
18.17.7.245 OFFSET 2324
18.17.7.246 OR 2325
18.17.7.247 PEARSON 2326
18.17.7.248 PERCENTILE 2327
18.17.7.249 PERCENTRANK 2327
18.17.7.250 PERMUT 2328
18.17.7.251 PHONETIC 2329
18.17.7.252 PI 2329
18.17.7.253 PMT 2330
18.17.7.254 POISSON 2331
18.17.7.255 POWER 2332
18.17.7.256 PPMT 2333
18.17.7.257 PRICE 2334
18.17.7.258 PRICEDISC 2337
18.17.7.259 PRICEMAT 2340
18.17.7.260 PROB 2342
18.17.7.261 PRODUCT 2343
18.17.7.262 PROPER 2344
18.17.7.263 PV 2344
18.17.7.264 QUARTILE 2346
18.17.7.265 QUOTIENT 2347
18.17.7.266 RADIANS 2347
18.17.7.267 RAND 2348
18.17.7.268 RANDBETWEEN 2348
18.17.7.269 RANK 2349
18.17.7.270 RATE 2349
18.17.7.271 RECEIVED 2351
18.17.7.272 REPLACE 2353
18.17.7.273 REPLACEB 2354
18.17.7.274 REPT 2355
18.17.7.275 RIGHT 2356
18.17.7.276 RIGHTB 2356
18.17.7.277 ROMAN 2357
18.17.7.278 ROUND 2359
18.17.7.279 ROUNDDOWN 2359
18.17.7.280 ROUNDUP 2360
18.17.7.281 ROW 2361
18.17.7.282 ROWS 2361

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.283 RSQ 2362
18.17.7.284 RTD 2363
18.17.7.285 SEARCH 2364
18.17.7.286 SEARCHB 2365
18.17.7.287 SECOND 2366
18.17.7.288 SERIESSUM 2367
18.17.7.289 SIGN 2368
18.17.7.290 SIN 2368
18.17.7.291 SINH 2369
18.17.7.292 SKEW 2369
18.17.7.293 SLN 2370
18.17.7.294 SLOPE 2371
18.17.7.295 SMALL 2372
18.17.7.296 SQRT 2373
18.17.7.297 SQRTPI 2373
18.17.7.298 STANDARDIZE 2374
18.17.7.299 STDEV 2375
18.17.7.300 STDEVA 2376
18.17.7.301 STDEVP 2376
18.17.7.302 STDEVPA 2377
18.17.7.303 STEYX 2379
18.17.7.304 SUBSTITUTE 2380
18.17.7.305 SUBTOTAL 2380
18.17.7.306 SUM 2382
18.17.7.307 SUMIF 2382
18.17.7.308 SUMIFS 2383
18.17.7.309 SUMPRODUCT 2385
18.17.7.310 SUMSQ 2385
18.17.7.311 SUMX2MY2 2386
18.17.7.312 SUMX2PY2 2387
18.17.7.313 SUMXMY2 2388
18.17.7.314 SYD 2389
18.17.7.315 T 2389
18.17.7.316 TAN 2390
18.17.7.317 TANH 2390
18.17.7.318 TBILLEQ 2391
18.17.7.319 TBILLPRICE 2392
18.17.7.320 TBILLYIELD 2392
18.17.7.321 TDIST 2393
18.17.7.322 TEXT 2394
18.17.7.323 TIME 2395
18.17.7.324 TIMEVALUE 2396
18.17.7.325 TINV 2396
18.17.7.326 TODAY 2397
18.17.7.327 TRANSPOSE 2397
18.17.7.328 TREND 2398
18.17.7.329 TRIM 2399
18.17.7.330 TRIMMEAN 2399

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.17.7.331 TRUE 2400
18.17.7.332 TRUNC 2400
18.17.7.333 TTEST 2401
18.17.7.334 TYPE 2402
18.17.7.335 UPPER 2403
18.17.7.336 USDOLLAR 2404
18.17.7.337 VALUE 2404
18.17.7.338 VAR 2405
18.17.7.339 VARA 2406
18.17.7.340 VARP 2407
18.17.7.341 VARPA 2408
18.17.7.342 VDB 2409
18.17.7.343 VLOOKUP 2410
18.17.7.344 WEEKDAY 2412
18.17.7.345 WEEKNUM 2413
18.17.7.346 WEIBULL 2414
18.17.7.347 WORKDAY 2415
18.17.7.348 WORKDAY.INTL 2416
18.17.7.349 XIRR 2418
18.17.7.350 XNPV 2420
18.17.7.351 YEAR 2421
18.17.7.352 YEARFRAC 2421
18.17.7.353 YIELD 2424
18.17.7.354 YIELDDISC 2428
18.17.7.355 YIELDMAT 2430
18.17.7.356 ZTEST 2433

18.18 Simple Types 2434

18.18.1 ST_Axis (PivotTable Axis) 2434
18.18.2 ST_BorderId (Border Id) 2434
18.18.3 ST_BorderStyle (Border Line Styles) 2434
18.18.4 ST_CalcMode (Calculation Mode) 2437
18.18.5 ST_CellComments (Cell Comments) 2438
18.18.6 ST_CellFormulaType (Formula Type) 2438
18.18.7 ST_CellRef (Cell Reference) 2439
18.18.8 ST_CellSpan (Cell Span Type) 2439
18.18.9 ST_CellSpans (Cell Spans) 2439
18.18.10 ST_CellStyleXfId (Cell Style Format Id) 2439
18.18.11 ST_CellType (Cell Type) 2439
18.18.12 ST_CfType (Conditional Format Type) 2440
18.18.13 ST_CfvoType (Conditional Format Value Object Type) 2442
18.18.14 ST_Comments (Comment Display Types) 2442
18.18.15 ST_ConditionalFormattingOperator (Conditional Format Operators) 2443
18.18.16 ST_CredMethod (Credentials Method) 2443
18.18.17 ST_DataConsolidateFunction (Data Consolidation Functions) 2444
18.18.18 ST_DataValidationErrorStyle (Data Validation Error Styles) 2444
18.18.19 ST_DataValidationImeMode (Data Validation IME Mode) 2445
18.18.20 ST_DataValidationOperator (Data Validation Operator) 2446

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.18.21 ST_DataValidationType (Data Validation Type) 2447
18.18.22 ST_DateTimeGrouping (Date Time Grouping) 2447
18.18.23 ST_DdeValueType (DDE Value Types) 2448
18.18.24 ST_DvAspect (Data View Aspect Type) 2448
18.18.25 ST_DxfId (Format Id) 2449
18.18.26 ST_DynamicFilterType (Dynamic Filter) 2449
18.18.27 ST_ExternalConnectionType (Text Field Datatype) 2451
18.18.28 ST_FieldSortType (Field Sort Type) 2451
18.18.29 ST_FileType (File Type) 2452
18.18.30 ST_FillId (Fill Id) 2452
18.18.31 ST_FilterOperator (Filter Operator) 2452
18.18.32 ST_FontId (Font Id) 2453
18.18.33 ST_FontScheme (Font scheme Styles) 2453
18.18.34 ST_FormatAction (PivotTable Format Types) 2453
18.18.35 ST_Formula (Formula) 2454
18.18.36 ST_FormulaExpression (Formula Expression Type) 2454
18.18.37 ST_GradientType (Gradient Type) 2454
18.18.38 ST_GroupBy (Values Group By) 2455
18.18.39 ST_GrowShrinkType (Grow Shrink Type) 2455
18.18.40 ST_HorizontalAlignment (Horizontal Alignment Type) 2456
18.18.41 ST_HtmlFmt (HTML Formatting Handling) 2460
18.18.42 ST_IconSetType (Icon Set Type) 2460
18.18.43 ST_ItemType (PivotItem Type) 2463
18.18.44 ST_MdxFunctionType (MDX Function Type) 2464
18.18.45 ST_MdxKPIProperty (MDX KPI Property) 2464
18.18.46 ST_MdxSetOrder (MDX Set Order) 2465
18.18.47 ST_NumFmtId (Number Format Id) 2465
18.18.48 ST_Objects (Object Display Types) 2465
18.18.49 ST_OleUpdate (OLE Update Types) 2466
18.18.50 ST_Orientation (Orientation) 2466
18.18.51 ST_PageOrder (Page Order) 2467
18.18.52 ST_Pane (Pane Types) 2467
18.18.53 ST_PaneState (Pane State) 2468
18.18.54 ST_ParameterType (Parameter Type) 2468
18.18.55 ST_PatternType (Pattern Type) 2469
18.18.56 ST_PhoneticAlignment (Phonetic Alignment Types) 2474
18.18.57 ST_PhoneticType (Phonetic Type) 2474
18.18.58 ST_PivotAreaType (Rule Type) 2475
18.18.59 ST_PivotFilterType (Pivot Filter Types) 2475
18.18.60 ST_PrintError (Print Errors) 2478
18.18.61 ST_Qualifier (Qualifier) 2478
18.18.62 ST_Ref (Cell References) 2479
18.18.63 ST_RefA (Single Cell Reference) 2479
18.18.64 ST_RefMode (Reference Mode) 2479
18.18.65 ST_RevisionAction (Revision Action Types) 2480
18.18.66 ST_rwColActionType (Row Column Action Type) 2480
18.18.67 ST_Scope (Conditional Formatting Scope) 2481
18.18.68 ST_SheetState (Sheet Visibility Types) 2481

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.18.69 ST_SheetViewType (Sheet View Type) 2481
18.18.70 ST_ShowDataAs (Show Data As) 2482
18.18.71 ST_SmartTagShow (Smart Tag Display Types) 2483
18.18.72 ST_SortBy (Sort By) 2483
18.18.73 ST_SortMethod (Sort Method) 2484
18.18.74 ST_SortType (Set Sort Order) 2484
18.18.75 ST_SourceType (PivotCache Type) 2485
18.18.76 ST_Sqref (Reference Sequence) 2485
18.18.77 ST_TableStyleType (Table Style Type) 2485
18.18.78 ST_TableType (Table Type) 2504
18.18.79 ST_TargetScreenSize (Target Screen Size Types) 2504
18.18.80 ST_TextHAlign (Comment Text Horizontal Alignment) 2505
18.18.81 ST_TextVAlign (Comment Text Vertical Alignment) 2505
18.18.82 ST_TimePeriod (Time Period Types) 2506
18.18.83 ST_TotalsRowFunction (Totals Row Function Types) 2506
18.18.84 ST_Type (Top N Evaluation Type) 2507
18.18.85 ST_UnderlineValues (Underline Types) 2507
18.18.86 ST_UnsignedIntHex (Hex Unsigned Integer) 2508
18.18.87 ST_UpdateLinks (Update Links Behavior Types) 2508
18.18.88 ST_VerticalAlignment (Vertical Alignment Types) 2509
18.18.89 ST_Visibility (Visibility Types) 2512
18.18.90 ST_VolDepType (Volatile Dependency Types) 2512
18.18.91 ST_VolValueType (Volatile Dependency Value Types) 2513
18.18.92 ST_WebSourceType (Web Source Type) 2513
18.18.93 ST_XmlDataType (XML Data Types) 2514
18.18.94 ST_FontFamily (Font Family) 2514

End of informative text.

18.2 Workbook

A workbook is composed of workbook-level properties and a collection of 1 or more sheets. The sheets are the central structure within a workbook, and can contain cells, which, in turn, can contain the text, numbers, dates, formulas, and other constructs of a workbook. The workbook part and corresponding properties comprise data used to set application and workbook-level operational state. The workbook also serves to bind all the sheets and child elements into an organized single file. The workbook XML attributes and elements include information about what application last saved the file, where and how the windows of the workbook were positioned, and an enumeration of the worksheets in the workbook.

It is important for the sake of simplicity to minimize the *required* set of workbook XML attributes and elements that shall be present to compose a valid SpreadsheetML workbook. Therefore this is the XML for the smallest possible (blank) workbook:

```
<workbook>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
  <sheets>
    <sheet name="Sheet1" sheetId="1" r:id="rId1"/>
  </sheets>
</workbook>
```

Note that this workbook has a single sheet, named Sheet1. An Id for the sheet is required, and a relationship Id pointing to the location of the sheet definition is also required.

18.2.1 bookViews (Workbook Views)

This element specifies the collection of workbook views of the enclosing workbook. Each view can specify a window position, filter options, and other configurations. There is no limit on the number of workbook views that can be defined for a workbook.

[*Example*:

```
<bookViews>
  <workbookView showHorizontalScroll="0" showVerticalScroll="0"
    showSheetTabs="0" xWindow="120" yWindow="45" windowWidth="15135"
    windowHeight="8130" activeTab="2" autoFilterDateGrouping="0"/>
</bookViews>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BookViews) is located in §A.2. *end note*]

18.2.2 calcPr (Calculation Properties)

This element defines the collection of properties the application uses to record calculation status and details. Calculation is the process of computing formulas and then displaying the results as values in the cells that contain the formulas.

[*Example*:

```
<calcPr calcId="122211" calcMode="auto" refMode="R1C1" iterate="1"
  fullPrecision="0"/>
```

end example]

Attributes	Description
calcCompleted (Calc Completed)	Specifies a boolean value that determines whether workbook data was recalculated before the workbook was saved. A value of `1` or `true` indicates recalculation was completed before save. A value of `0` or `false` indicates that recalculation was not completed before save.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
calcId (Calculation Id)	Specifies the version of the calculation engine used to calculate values in the workbook. When you open a workbook created in the current version, the application recalculates only the formulas that depend on cells that have changed. When you open a workbook that was created in an earlier version of the application, all the formulas in the workbook— those that depend on cells that have changed and those that do not— are recalculated. This ensures that the workbook is fully optimized for the current application version. The value for `calcID` depends on the application. SpreadsheetML defaults form [version][build], where [version] refers to the version of the application, and [build] refers to the build of the application when the calculation engine changed. [*Example*: `<calcPr calcId="122211"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
calcMode (Calculation Mode)	Specifies when the application should calculate formulas in the workbook. The default value for this attribute is "auto." The possible values for this attribute are defined by the ST_CalcMode simple type (§18.18.4).
calcOnSave (Calculate On Save)	Specifies a boolean value that indicates whether the application recalculates values when the workbook is saved. A value of `1` or `true` indicates recalculation is performed when the workbook is saved. A value of `0` or `false` indicates recalculation is not performed when the workbook is saved. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
concurrentCalc (Concurrent	Specifies a boolean value that indicates whether concurrent calculation processes are enabled for this workbook.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Calculations)	A value of on, 1, or true indicates concurrent calculations are enabled in this workbook. A value of `0` or `false` indicates concurrent calculations are not enabled. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
concurrentManualCount (Concurrent Thread Manual Count)	Specifies the count of concurrent calculation processes manually set by the user. If omitted, the count is set automatically by the application. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
forceFullCalc (Force Full Calculation)	Specifies a boolean value that indicates whether the application performs a full recalculation when one was not indicated by other calculation properties. This attribute allows the application to expose mechanisms in the user interface that give users the ability to trigger when full recalculations take place. A value of `1` or `true` indicates the application performs a full recalculation of the workbook. A value of `0` or `false` indicates the application does not perform a full recalculation of the workbook. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fullCalcOnLoad (Full Calculation On Load)	Specifies a boolean value that indicates whether the application shall perform a full recalculation when the workbook is opened. After load and successful calculation, the application should set this value to false. The application should set this value to true when cell formulas or values are modified by another process while the application has the workbook opened. A value of `1` or `true` indicates the application performs a full recalculation of workbook values when the workbook is opened. A value of `0` or `false` indicates the application does not perform a full recalculation when the workbook is opened. [*Note*: If manual calcMode is `true`, then a full recalculation does not performed on load, even when this attribute is set. *end note*] The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
fullPrecision (Full Precision Calculation)	Specifies a boolean that indicates the precision the application uses when performing calculations in the workbook. Full precision means that the application uses the entire value(s) stored in cells referenced by the formula to perform the calculation. [*Example*: If two cells each contain the value 10.005 and the cells are formatted to display values in currency format, the value $10.01 is displayed in each cell. If you add the two cells together, the result is $20.01 because the application adds the stored values 10.005 and 10.005, not the displayed values. You can change the precision of calculations so that the application uses the displayed value instead of the stored value when it recalculates formulas. For the above example, if fullPrecision is `false`, then the result must be $20.02, because each cell shows $10.01, so those are the values to be added. Furthermore, when fullPrecision is `false`, the calculated value as displayed must be saved to file. *end example*] A value of `1` or `true` indicates the application uses the stored values of the referenced cells when performing calculations. A value of `0` or `false` indicates the application uses the display values of the referenced cells when performing calculations. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
iterate (Calculation Iteration)	Specifies a boolean value that indicates whether the application should attempt to calculate formulas that contain circular references. A circular reference is a formula that refers to the cell— either directly or indirectly— that contains the formula. If a formula refers back to one of its own cells, you shall determine how many times the formula should recalculate. A value of on, 1, or true indicates the application should attempt to calculate circular references. The calculation engine performs iterative `iterateCount` calculations to before stopping. A value of `0` or `false` indicates that the application should not attempt to calculate formulas with circular references. The calculation engine stops on the first iteration when it encounters a circular references. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
iterateCount (Iteration Count)	Specifies the number of iterations the calculation engine attempts when calculating a workbook with circular references, when `iterate` is true.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The default value for this attribute is 100. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
iterateDelta (Iterative Calculation Delta)	Specifies a double that contains the maximum change for iterative calculations. The application stops calculating after iterateCount iterations or after all values in the circular reference change by less than iterateDelta between iterations, whichever comes first. The default value for this attribute is "0.001" The possible values for this attribute are defined by the W3C XML Schema double datatype.
refMode (Reference Mode)	Specifies the reference style for this workbook. Instead of using letters for columns and numbers for rows ("A1"), this option enables using numbers for both rows and columns. Cells are then referred to in this format: R1C1. The default value for this attribute is "A1." The possible values for this attribute are defined by the ST_RefMode simple type (§18.18.64).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalcPr) is located in §A.2. *end note*]

18.2.3 customWorkbookView (Custom Workbook View)

This element specifies a single custom workbook view. A custom workbook view consists of a set of display and print settings that you can name and apply to a workbook. You can create more than one custom workbook view of the same workbook. Custom Workbook Views are not required in order to construct a valid SpreadsheetML document, and are not necessary if the document is never displayed by a spreadsheet application, or if the spreadsheet application has a fixed display for workbooks. However, if a spreadsheet application chooses to implement configurable display modes, the customWorkbookView element should be used to persist the settings for those display modes. The settings fall into the following categories, and individual settings are detailed in the table following:

Window settings - these include size and positioning of the spreadsheet window as well as which window features should be displayed (scrollbars, sheet tabs, etc.).

Workbook content display and print settings - specifies whether features in the workbook should be included or ignored by a spreadsheet application when a custom workbook view is displayed or printed. [*Example*: Whether comments in the workbook should be displayed and how objects such as images should be displayed can be controlled by a Custom Workbook View. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Persistence settings - these include settings that describe how a spreadsheet application should update persisted SpreadsheetML content if multiple spreadsheet applications are accessing a common instance of the SpreadsheetML document at the same time.

When a Custom Workbook View is present, there should also be corresponding customSheetView (§18.3.1.25) elements for each sheet (§18.2.19) in the workbook. The guid attribute of these customSheetView elements associates the customSheetView with the appropriate customWorkbookView. Attributes on the customWorkbookView element should be used to determine which settings within the Custom Sheet View should be respected by a spreadsheet application if the Custom Workbook View is displayed.

Attributes	Description
activeSheetId (Active Sheet in Book View)	Specifies the sheetId of a sheet in the workbook that identifies to a consuming application the default sheet to display. Corresponds to a sheetId of a sheet in the sheets collection. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
autoUpdate (Auto Update)	Specifies a boolean value that is an instruction that if the workbook is loaded by a spreadsheet application, that spreadsheet application should automatically update changes at the interval specified by the `mergeInterval` attribute. This is only applicable for shared workbooks (§18.11). A value of `1` or `true` is an instruction to the spreadsheet application to update changes at the interval specified in the `mergeInterval` attribute. A value of `0` or `false` is an instruction to the spreadsheet applicationto update changes whenever the spreadsheet application generates SpreadsheetML representing the workbook. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
changesSavedWin (Changes Saved Win)	Specifies a boolean value that instructs a spreadsheet application to overwrite the persisted version of the document with the updated version being persisted. This is only applicable for shared workbooks in automatic update mode. A value of `1` or `true` instructs a spreadsheet application to overwrite changes in the persisted version of a shared workbook when conflicts in data are found. A value of `0` or `false` instructs a spreadsheet application to not overwrite changes in the persisted version of a shared workbook when conflicts are found. The default value for this attribute is false.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
guid (Custom View GUID)	Specifies a globally unique identifier (GUID) for this custom view The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
includeHiddenRowCol (Include Hidden Rows & Columns)	Specifies a boolean value that indicates whether to include hidden rows, columns, and filter settings in this custom view. A value of `1` or `true` indicates that hidden rows, columns, and filter settings are included in this custom view. A value of `0` or `false` indicates that hidden rows, columns, and filter settings are not included. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
includePrintSettings (Include Print Settings)	Specifies a boolean value that indicates whether to include print settings in this custom view. A value of `1` or `true` indicates that print settings are included in this custom view. A value of `0` or `false` indicates print settings are not included in this custom view. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
maximized (Maximized)	Specifies a boolean value that indicates whether the workbook window is maximized. A value of `1` or `true` indicates the workbook window is maximized. A value of `0` or `false` indicates the workbook window is not maximized. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
mergeInterval (Merge Interval)	Automatic update interval (in minutes). Only applicable for shared workbooks in automatic update mode. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
minimized (Minimized)	Specifies a boolean value that indicates whether the workbook window is minimized. A value of `1` or `true` indicates the workbook window is minimized. A value of `0` or `false` indicates the workbook window is not minimized. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Custom View Name)	Specifies the name of the custom view. This attribute is required. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
onlySync (Only Synch)	Specifies a boolean value that indicates, during automatic update, the current user's changes are not saved. The workbook is only updated with other users' changes. Only applicable for shared workbooks in automatic update mode. A value of `1` or `true` indicates the current user's changes is not saved during automatic update. A value of `0` or `false` indicates the current user's is saved during automatic update. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
personalView (Personal View)	Specifies a boolean value that indicates that this custom view is a personal view for a shared workbook user. Only applicable for shared workbooks. Personal views allow each user of a shared workbook to store their individual print and filter settings. A value of `1` or `true` indicates this custom view is a personal view for a shared workbook user. A value of `0` or `false` indicates this view is not a personal view. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showComments	Specifies how comments are displayed in this custom view

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Show Comments)	The possible values for this attribute are defined by the ST_Comments simple type (§18.18.14).
showFormulaBar (Show Formula Bar)	Specifies a boolean value that indicates whether to display the formula bar in the application user interface. A value of `1` or `true` indicates the formula bar is shown in the user interface. A value of `0` or `false` indicates the formula bar is not shown in the user interface. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showHorizontalScroll (Show Horizontal Scroll)	Specifies a boolean value that indicates whether to display the horizontal scroll bar in the user interface. A value of `1` or `true` indicates that the horizontal scrollbar is shown. A value of `0` or `false` indicates that the horizontal scrollbar is not shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showObjects (Show Objects)	Specifies how objects are displayed in this custom view. The default value for this attribute is "all." The possible values for this attribute are defined by the ST_Objects simple type (§18.18.48).
showSheetTabs (Show Sheet Tabs)	Specifies a boolean value that indicates whether to display the sheet tabs in the user interface. A value of `1` or `true` indicates that sheet tabs shall be shown. A value of `0` or `false` indicates that sheet tabs shall not be shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showStatusbar (Show Status Bar)	Specifies a boolean value that indicates whether to display the status bar in the user interface.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `1` or `true` indicates that the status bar is shown. A value of `0` or `false` indicates the status bar is not shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showVerticalScroll (Show Vertical Scroll)	Specifies a boolean value that indicates whether to display the vertical scroll bar. A value of `1` or `true` indicates the vertical scrollbar shall be shown. A value of `0` or `false` indicates the vertical scrollbar shall not be shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tabRatio (Sheet Tab Ratio)	Specifies the ratio between the workbook tabs bar and the horizontal scroll bar. `tabRatio` is assumed to be out of 1000 of the horizontal window width. The default value for this attribute is 600. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
windowHeight (Window Height)	Specifies the height of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
windowWidth (Window Width)	Specifies the width of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xWindow (Top Left Corner (X Coordinate))	Specifies the X coordinate for the upper left corner of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema int datatype.
yWindow (Top Left Corner (Y Coordinate))	Specifies the Y coordinate for the upper left corner of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema int datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomWorkbookView) is located in §A.2. *end note*]

18.2.4 customWorkbookViews (Custom Workbook Views)

This element defines the collection of custom workbook views that are defined for this workbook. A customWorkbookView is similar in concept to a workbookView (§18.2.30) in that its attributes contain settings related to the way that the workbook should be displayed on a screen by a spreadsheet application.

[*Note*: Whilst a workbookView is intended to store the way the workbook window should be displayed by a spreadsheet application, customWorkbookView elements are intended to allow the user to switch between a selection of customWorkbookView items defining window and content display options. *end note*]

A customWorkbookView contains a greater number of settings (e.g., the presence of a formula bar; visibility of hidden data; whether or not to show comments) and is named.

[*Example*: A workbook which is used by two different departments might contain two customWorkbookView elements – one where the comments and hidden data are not shown, and one where they are. Users might switch between the customWorkbookView items according to the department in which they work. *end example*]

There is no limit on the number of custom views that can be contained within a SpreadsheetML instance.

[*Example*:

```
<customWorkbookViews>
  <customWorkbookView name="CustomView"
    guid="{CE6681F1-E999-414D-8446-68A031534B57}" maximized="1" xWindow="1"
    yWindow="1" windowWidth="1024" windowHeight="547" activeSheetId="1"/>
</customWorkbookViews>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomWorkbookViews) is located in §A.2. *end note*]

18.2.5 definedName (Defined Name)

This element defines a defined name within this workbook. A defined name is descriptive text that is used to represents a cell, range of cells, formula, or constant value.

[*Example*: A defined name can make it easier to refer to ranges. The name Products might be easier to understand than the ranges Sales!C20:C30. *end example*]
[*Example*: A defined name in a formula can make it easier to understand the purpose of the formula. For example, the formula =SUM(FirstQuarterSales) might be easier to identify than =SUM(C20:C30). *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Names are available to any sheet. For example, if the name ProjectedSales refers to the range A20:A30 on the first worksheet in a workbook, the name ProjectedSales can be used on any other sheet in the same workbook to refer to range A20:A30 on the first worksheet. *end example*]

[*Example*: Names can be used to represent formulas or values that do not change (constants). For example, the name SalesTax can be used to represent the sales tax amount (such as 6.2 percent) applied to sales transactions. *end example*]

[*Example*: A defined name in another workbook may be referenced and a defined name may refer to cells in another workbook. For example, the formula =SUM(Sales.xls!ProjectedSales) may refer to the named range ProjectedSales in the workbook named Sales. *end example*]

A compliant producer or consumer considers a defined name in the range A1-XFD1048576 to be an error.

All other names outside this range can be defined as names and overrides a cell reference if an ambiguity exists.

[*Example*: For clarification: LOG10 is always a cell reference, LOG10() is always a formula, LOGO1000 can be a defined name that overrides a cell reference. *end example*]

Attributes	Description
comment (Comment)	Specifies the comment the user provided when the name was created. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
customMenu (Custom Menu Text)	Specifies custom menu text for the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
description (Description)	Specifies description text for the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
function (Function)	Specifies a boolean value that indicates that the defined name refers to a user-defined function. This attribute is used when there is an add-in or other code project associated with the file. A value of `1` or `true` indicates the name refers to a function. A value of `0` or `false` indicates the name does not refer to a function. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
functionGroupId	Specifies the function group index if the defined name refers to a function. The function

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Function Group Id)	group defines the general category for the function. This attribute is used when there is an add-in or other code project associated with the file. The following functionGroupIds are defined in SpreadsheetML for applications that support the association of an add-in or code project for their workbook: • 1 Financial • 2 Date and Time • 3 Math and Trig • 4 Statistical • 5 Lookup and Reference • 6 Database • 7 Text • 8 Logical • 9 Information • 10 Commands • 11 Customizing • 12 Macro Control • 13 DDE / External • 14 User Defined • 15 Engineering • 16 Cube The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
help (Help)	Specifies the help topic to display for this defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
hidden (Hidden Name)	Specifies a boolean value that indicates whether the defined name is hidden in the user interface. A value of `1` or `true` indicates the name is hidden. A value of `0` or `false` indicates the name is not hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
localSheetId (Local Name Sheet Id)	Specifies the sheet index in this workbook where data from an external reference is displayed. [*Example*: In the following example, the defined name refers to a range whose data source is an external database called “Northwind_Database”: `<definedName name="Northwind_Database"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	`localSheetId="2">Sheet5!$A$1:$T$47</definedName>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Defined Name)	Specifies the name that appears in the user interface for the defined name. This attribute is required. The following built-in names are defined in this SpreadsheetML specification: • • **Print** • _xlnm .Print_Area: this defined name specifies the workbook's print area. • _xlnm .Print_Titles: this defined name specifies the row(s) or column(s) to repeat at the top of each printed page. • • **Filter & Advanced Filter** • _xlnm .Criteria: this defined name refers to a range containing the criteria values to be used in applying an advanced filter to a range of data. • _xlnm ._FilterDatabase: can be one of the following a. this defined name refers to a range to which an advanced filter has been applied. This represents the source data range, unfiltered. b. This defined name refers to a range to which an AutoFilter has been applied. • _xlnm .Extract: this defined name refers to the range containing the filtered output values resulting from applying an advanced filter criteria to a source range. • • **Miscellaneous** • _xlnm .Consolidate_Area: the defined name refers to a consolidation area. • _xlnm .Database: the range specified in the defined name is from a database data source. • _xlnm .Sheet_Title: the defined name refers to a sheet title. • Built-in names reserved by SpreadsheetML begin with "_xlnm.". End users shall not use this string for custom names in the user interface. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
publishToServer (Publish To Server)	Specifies a boolean value that indicates whether the defined name is included in the version of the workbook that is published to or rendered on a Web or application server. A value of `1` or `true` indicates the name shall be published. A value of `0` or `false` indicates the name shall not be published. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
shortcutKey (Shortcut Key)	Specifies the keyboard shortcut for the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
statusBar (Status Bar)	Specifies text that is displayed on the application status bar when the user places focus on the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
vbProcedure (Procedure)	Specifies a boolean value that indicates whether the defined name is related to an external function, command, or other executable code. A value of `1` or `true` indicates the name is related to an external function, command, or other executable code, and the loading application can optionally decide whether to load and/or execute the commands. A value of `0` or `false` indicates the name does not refer to an external function, command, or other executable code. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
workbookParameter (Workbook Parameter (Server))	Specifies a boolean value that indicates that the name is used as a workbook parameter on a version of the workbook that is published to or rendered on a Web or application server. A value of `1` or `true` indicates the name is used as a workbook parameter on the application server. A value of `0` or `false` indicates the name is not used as a workbook parameter on the application server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xlm (External Function)	Specifies a boolean value that indicates whether the defined name is related to an external function, command, or other executable code. A value of `1` or `true` indicates the name is related to an external function, command, or other executable code, and the loading application can optionally decide whether to load and/or execute the commands. A value of `0` or `false` indicates the name does not refer to an external function,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	command, or other executable code. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
Xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefinedName) is located in §A.2. *end note*]

18.2.6 definedNames (Defined Names)

This element defines the collection of defined names for this workbook. Defined names are descriptive names to represent cells, ranges of cells, formulas, or constant values. Defined names can be used to represent a range on any worksheet.

[*Example*:

```
<definedNames>
  <definedName name="NamedFormula"
    comment="Comment text for defined name.">SUM(Sheet3!$B$2:$B$9)</definedName>
  <definedName name="NamedRange">Sheet3!$A$1:$C$12</definedName>
  <definedName name="NamedRangeFromExternalReference" localSheetId="2"
    hidden="1">Sheet5!$A$1:$T$47</definedName>
</definedNames>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefinedNames) is located in §A.2. *end note*]

18.2.7 ext (Extension)

Each extension within an extension list shall be contained within an ext element. Extensions shall be versioned by namespace, using the uri attribute, and shall be allowed to appear in any order within the extension list. Any number of extensions shall be allowed within an extension list.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

When extension lists are processed, a consumer might understand some extensions, and might not understand other extensions. The preservation model for extensions is that unprocessed extensions shall always be preserved (when consuming) and written out (when producing) in whole, as long as there is not some ancestor element of the extension list that is discarded as a result of MCE processing. [*Example*: If, when consumed by a SpreadsheetML editor, a sheet contains several extensions within an extension list, but the sheet no longer exists after editing, the extensions associated with that sheet are not written out in the resulting edited SpreadsheetML document. *end example*].

Markup namespaces within extensions shall not be required to be listed in the Ignorable Compatibility-Rule attribute. [*Note*: See §10 for additional discussion on Application-Defined Extension Elements and processing rules. *end note*]

[*Example*:

In this example, there are two extensions written. The first extension describes a new structure that might have been defined by a fictitious second version of ISO/IEC 29500. The second extension describes a structure that might be introduced by a private party, developed independently from ISO/IEC 29500.

```
<extLst>
  <ext uri='http://purl.oclc.org/ooxml/spreadsheetml/versionTwoExtension'>
    <v2:newContent
xmln:v2='http://purl.oclc.org/ooxml/spreadsheetml/versionTwoExtension'>
      …
    </v2:newContent>
  </ext>
  <ext uri='http://www.extension.com/versionOneExtension'>
    <v2:moreContent xmlsn:v2='http://www.extension.com/versionOneExtension'>
      …
    </v2:moreContent>
  </ext>
</extList>
```

end example]

Each extension has an uri attribute, which serves as an identifier to indicate information about the extension. [*Note*: For example, the uri might state the version of a markup specification to which the content conforms, or it might state the version of a producing application that wrote the content. *end note*] Upon encountering extensions, a processing consumer shall determine whether it knows how to process extensions using the value of the uri. If the consumer knows how to process such an extension, the markup contained within that extension is processed. Otherwise, the extension content shall be preserved so long as the structure that contains the extLst has not been removed.

Attributes	Description
uri (URI)	A token to identify version and application information for this particular extension.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Extension) is located in §A.2. *end note*]

18.2.8 externalReference (External Reference)

This element defines an external reference that stores data for workbook elements.

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies a unique identifier that is used to identify a relationship to another part in the file. Relationship identifiers link the element definition with the part where data for the element is stored. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalReference) is located in §A.2. *end note*]

18.2.9 externalReferences (External References)

This element defines the collection of external references for this workbook.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalReferences) is located in §A.2. *end note*]

18.2.10 extLst (Future Feature Data Storage Area)

This element provides a convention for extending spreadsheetML in predefined locations. The locations shall be denoted with the extLst element, and are called extension lists. Extension list locations within the markup document are specified in the markup specification and can be used to store extensions to the markup specification, whether those are future version extensions of the markup specification or are private extensions implemented independently from the markup specification. Markup within an extension might not be understood by a consumer.

extLst elements contain ext elements, called *extensions*. See §18.2.7 for more discussion of extensions.

[*Note*: Allowing markup specification extensions and private markup extensions within an extension list does not violate interoperability because the rules articulated within §10, §18.2.7 and Part 3 describe how producers and

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

consumers must generate and consume markup documents containing application defined extension elements. *end note*]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExtensionList) is located in §A.2. *end note*]

18.2.11 fileRecoveryPr (File Recovery Properties)

This element defines properties that track the state of the workbook file, such as whether the file was saved during a crash, or whether it should be opened in auto-recover mode.

Attributes	Description
autoRecover (Auto Recover)	Specifies a boolean value that indicates whether the file is mark for auto-recovery. Applications typically mark files for auto-recover following a crash. A value of `1` or `true` indicates the file is marked for auto-recover. A value of `0` or `false` indicates the file is not marked for auto-recover. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
crashSave (Crash Save)	Specifies a boolean value that indicates whether the application last saved the workbook file after a crash. A value of `1` or `true` indicates the workbook was last saved after a crash. A value of `0` or `false` indicates was not last saved as part of a crash. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataExtractLoad (Data Extract Load)	Specifies a boolean value that indicates whether the application last opened the workbook for data recovery. A value of `1` or `true` indicates the workbook was last opened for data recovery. A value of `0` or `false` indicates was not last opened for data recovery. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
repairLoad (Repair Load)	Specifies a boolean value that indicates whether the application last opened the workbook in safe or repair mode. A value of `1` or `true` indicates the workbook was last opened in safe or repair mode. A value of `0` or `false` indicates the workbook was last opened without problems. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FileRecoveryPr) is located in §A.2. *end note*]

18.2.12 fileSharing (File Sharing)

This element tracks file sharing settings for the workbook. When a password is to be hashed and stored in this element, it shall be hashed starting from a UTF-16LE encoded string value. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

Attributes	Description
algorithmName (Cryptographic Algorithm Name)	Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value. The following values are reserved:

Value	Algorithm
`MD2`	Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]
`MD4`	Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]
`MD5`	Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>

[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:

<pre>< … algorithmName="SHA-1"
  hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The algorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. end example]

The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
</td></tr>
<tr><td>hashValue
(Password Hash Value)</td><td>Specifies the hash value for the password required for editing this workbook. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.

If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.

[Example: Consider an Office Open XML document with the following information stored</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	in one of its protection elements:
	
	`<… algorithmName="SHA-1"`
	`hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hashValue attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the cryptAlgorithmSid attribute value of 1) and that the resulting has value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

readOnlyRecommended (Read Only Recommended)	Specifies a boolean value that indicates on open, whether the application alerts the user that the file be marked as read-only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

saltValue (Salt Value for Password Verifier)	Specifies the salt that was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and that shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack").
	
	If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value.
	
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements:
	
	`<… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ=="`
	`hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The saltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The spinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
userName (User Name)	Specifies the username of the person with write reservation for this workbook. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FileSharing) is located in §A.2. *end note*]

18.2.13 fileVersion (File Version)

This element defines properties that track which version of the application accessed the data and source code contained in the file.

Attributes	Description
appName (Application Name)	Specifies the application name. When saving, applications can write their appName value and optionally write `lastEdited` and `lowestEdited` attributes to track the version of the application that performed those actions. When opening the workbook, applications can examine the value of `appName` and decide how to interpret the `lastEdited`, `lowestEdited`, and `rupBuild` attributes. The possible values for this attribute are defined by the W3C XML Schema string datatype.
codeName (Code Name)	Specifies the GUID that identifies the code project that is associated with the workbook. [*Note*: the primary use of this attribute is to track the version of the compiled code. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
lastEdited (Last Edited Version)	Specifies the version of the application that last saved the workbook. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema string datatype.
lowestEdited (Lowest Edited Version)	Specifies the earliest version of the application that saved the workbook. This value is reset any time an application that can read all data in the file saves the file. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema string datatype.
rupBuild (Build Version)	Specifies the incremental public release of the application. [*Example*: Betas, service packs, and versions. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FileVersion) is located in §A.2. *end note*]

18.2.14 functionGroup (Function Group)

This element represents a single function group.

Attributes	Description
name (Name)	Specifies the name of the function group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FunctionGroup) is located in §A.2. *end note*]

18.2.15 functionGroups (Function Groups)

This element defines the collection of function groups for the workbook.

Attributes	Description
builtInGroupCount (Built-in Function Group Count)	Specifies the count of built-in function groups that the application provides in this workbook.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The default value for this attribute is 16. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FunctionGroups) is located in §A.2. *end note*]

18.2.16 oleSize (Embedded Object Size)

This element defines the embedded object server for this workbook.

Attributes	Description
ref (Reference)	Specifies the reference for the embedded object. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleSize) is located in §A.2. *end note*]

18.2.17 pivotCache (PivotCache)

This element represents a cache of data for pivot tables and formulas in the workbook.

Attributes	Description
cacheId (PivotCache Id)	Specifies the unique identifier for the pivot cache for this workbook in the pivot cache part. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identifier to a pivot cache definition part where cached data is stored. This attribute is required. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotCache) is located in §A.2. *end note*]

18.2.18 pivotCaches (PivotCaches)

This element enumerates pivot cache definition parts used by pivot tables and formulas in this workbook.

[*Example*:

```
<pivotCaches>
  <pivotCache cacheId="4" r:id="rId8"/>
</pivotCaches>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotCaches) is located in §A.2. *end note*]

18.2.19 sheet (Sheet Information)

This element defines a sheet in this workbook. Sheet data is stored in a separate part.

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identifier of the sheet part where the definition for this sheet is stored. This attribute is required. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sheet Name)	Specifies the name of the sheet. This name shall be unique. This attribute is required. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sheetId (Sheet Tab Id)	Specifies the internal identifier for the sheet. This identifier shall be unique. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
state (Visible State)	Specifies the visible state of this sheet. The default value for this attribute is "visible."

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_SheetState simple type (§18.18.68).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sheet) is located in §A.2. *end note*]

18.2.20 sheets (Sheets)

This element represents the collection of sheets in the workbook. There are different types of sheets you can create in SpreadsheetML. The most common sheet type is a worksheet; also called a spreadsheet. A worksheet is the primary document that you use in SpreadsheetML to store and work with data. A worksheet consists of cells that are organized into columns and rows.

Some workbooks might have a modular design where there is one sheet for data and another worksheet for each specific analysis performed on that data. In a complex modular system, you might have dozens of sheets, each dedicated to a specific task.

[*Example*:

```
<sheets>
  <sheet name="Sheet1" sheetId="1" r:id="rId1"/>
  <sheet name="Sheet2" sheetId="2" r:id="rId2"/>
  <sheet name="Sheet5" sheetId="3" r:id="rId3"/>
  <sheet name="Chart1" sheetId="4" r:id="rId4"/>
</sheets>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sheets) is located in §A.2. *end note*]

18.2.21 smartTagPr (Smart Tag Properties)

This element defines a collection of smart tag properties that determine smart tag behavior in the workbook.

[*Example*:

```
<smartTagPr embed="1" show="noIndicator"/>
```

end example]

Attributes	Description
embed (Embed SmartTags)	Specifies a boolean value that indicates whether the application saves smart tags with the workbook. Smart tag information is saved both in the workbook part and the sheet parts. A value of 1 or `true` indicates the application saves smart tags with the workbook.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	A value of `0` or `false` indicates the application does not save smart tags with the workbook. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
show (Show Smart Tags)	Specifies how the application displays smart tags in the user interface. The default value for this attribute is "all." The possible values for this attribute are defined by the ST_SmartTagShow simple type (§18.18.71).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagPr) is located in §A.2. *end note*]

18.2.22 smartTagType (Smart Tag Type)

This element represents a smart tag in the workbook.

Attributes	Description
name (Name)	Specifies the element name used for a smart tag that is used by the application. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
namespaceUri (SmartTag Namespace URI)	Specifies the namespace Uniform Resource Identifier (URI) for a smart tag used by the application. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
url (Smart Tag URL)	Specifies the URL for a smart tag provided by the smart tag provider in the application. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagType) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.2.23 smartTagTypes (Smart Tag Types)

This element defines the collection of smart tag types in the workbook. Smart tags represent data that is recognized and labeled as a particular type.

[*Example*: For example, a person's name or a date can be recognized and labeled with a smart tag.

```
<smartTagTypes>
  <smartTagType namespaceUri="urn:schemas-openxmlformats-org:office:smarttags"
    name="date"/>
</smartTagTypes>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTagTypes) is located in §A.2. *end note*]

18.2.24 webPublishing (Web Publishing Properties)

This element defines properties that relate to publishing this workbook to the Web.

Attributes	Description
allowPng (Allow PNG)	Specifies a boolean value that indicates whether the application saves images in the PNG (Portable Network Graphics) graphic format. A value of `1` or `true` indicates the application supports PNG . A value of `0` or `false` indicates the application does not support PNG. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
characterSet (Character Set)	Name of the character set the application uses when a Web page is saved. The values allowed within this attribute are names and aliases listed in the IANA CHARACTER SETS listing found at http://www.iana.org/assignments/character-sets. If this attribute is not present then the codePage attribute can be used. The possible values for this attribute are defined by the W3C XML Schema string datatype.
css (Use CSS)	Specifies a boolean value that indicates whether the application uses Cascading Style Sheet (CSS) for font formatting on Web pages. A value of `1` or `true` indicates the application uses CSS for font formats in Web pages. A value of `0` or `false` indicates the application does not use CSS for font formats.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dpi (DPI)	Specifies the DPI (defined as the number of pixels per inch) that are used to display images in Web pages. The specified DPI affects the size of graphics relative to the size of text on the screen. The default value for this attribute is 96. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
longFileNames (Enable Long File Names)	Specifies a boolean value that indicates whether the application allows file names longer than 8 octets with a three octet extension for Web pages. File names are not case-sensitive. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
targetScreenSize (Target Screen Size)	Specifies the screen size on which Web pages are displayed. The specified screen size might affect the size and layout of images on web pages. The default value of this attribute is "800x600." The possible values for this attribute are defined by the ST_TargetScreenSize simple type (§18.18.79).
thicket (Thicket)	Specifies a boolean value that indicates that the application stores supporting files such as bullets, background textures, and graphics in a separate folder from the Web page The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vml (VML in Browsers)	Specifies a boolean value that indicates whether the application uses VML (Vector Markup Language) to display graphics in Web browsers The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPublishing) is located in §A.2. *end note*]

18.2.25 webPublishObject (Web Publishing Object)

This element defines a single Web publishing object for the workbook. This element tracks basic information about an object in the workbook, such as a named range, that is published to the Web.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
autoRepublish (Auto Republish)	Specifies a boolean value that indicates whether the object specified in `sourceObject` is automatically published every time the workbook is saved. A value of `1` or `true` indicates the application will publish the `sourceObject` when the workbook is saved. A value of `0` or `false` indicates the application will not publish the `sourceObject` when the workbook is saved. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
destinationFile (Destination File)	Specifies the destination file name to which the `sourceObject` is published. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
divId (Div Id)	Specifies the destination bookmark (div id) for the published object. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
id (Id)	Specifies the number, in "nnnnn" format, used in generated div id, in style id's, token filenames, and other variables. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sourceObject (Source Object)	Specifies the named range to be published. If omitted, the entire workbook is published. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
title (Title)	Specifies the title of the published item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPublishObject) is located in §A.2. *end note*]

18.2.26 webPublishObjects (Web Publish Objects)

This element defines the collection of Web publishing objects in the workbook.

Attributes	Description
count (Count)	Specifies the number of items in the collection.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPublishObjects) is located in §A.2. *end note*]

18.2.27 workbook (Workbook)

The workbook element is the top level element. It contains elements and attributes that encompass the data content of the workbook. The workbook's child elements each have their own subclause references, and these are shown in the child elements table below. A partial list of the workbook's structures that these elements represent are:

- Sheets: represents the collection of worksheets in the workbook. The sheets are the central structure within a workbook, and contain the text, numbers, dates, formulas, and other elements of a workbook.
- Views: SpreadsheetML defines a collection of Workbook views that define basic window dimensions and position of the workbook if it is ever displayed by a spreadsheet application. It also defines a collection of Custom Workbook Views that allows SpreadsheetML to describe one or more views of the data within a workbook.
- Properties: the workbook has several property collection that store basic workbook settings, such as the date system to use, file protection settings, calculation settings, and smart tag behaviors.
- Names: words or strings of characters that represent cells, ranges of cells, formulas, or constant values.

[*Example*:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<workbook xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main"
mlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships">
  <fileVersion lastEdited="4" lowestEdited="4" rupBuild="4017"/>
  <workbookPr vbName="ThisWorkbook" defaultThemeVersion="123820"/>
  <bookViews>
    <workbookView xWindow="120" yWindow="45" windowWidth="15135"
      windowHeight="7650" activeTab="4"/>
  </bookViews>
  <sheets>
    <sheet name="Sheet1" sheetId="1" r:id="rId1"/>
    <sheet name="Sheet2" sheetId="2" r:id="rId2"/>
    <sheet name="Sheet5" sheetId="3" r:id="rId3"/>
    <sheet name="Chart1" sheetId="4" r:id="rId4"/>
  </sheets>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
  <definedNames>
    <definedName name="MyDefinedName">Sheet3!$A$1:$C$12</definedName>
  </definedNames>
  <calcPr calcId="122211" calcMode="autoNoTable" refMode="R1C1" iterate="1"
    fullPrecision="0"/>
  <customWorkbookViews>
    <customWorkbookView name="CustomView1"
      guid="{CE6681F1-E999-414D-8446-68A031534B57}" maximized="1" xWindow="1"
      yWindow="1" windowWidth="1024" windowHeight="547" activeSheetId="1"/>
  </customWorkbookViews>
  <pivotCaches>
    <pivotCache cacheId="0" r:id="rId8"/>
  </pivotCaches>
  <smartTagPr embed="1" show="noIndicator"/>
  <smartTagTypes>
    <smartTagType namespaceUri="urn:schemas-openxmlformats-org:office:smarttags"
      name="date"/>
  </smartTagTypes>
  <webPublishing codePage="1252"/>
</workbook>
```

end example]

Attributes	Description

conformance (Document Conformance Class)	Specifies the conformance class (§2.1) to which the SpreadsheetML document conforms.
	
	If this attribute is omitted, its default value is `transitional`.
	
	[*Example*: Consider the following SpreadsheetML Workbook part markup:
	
	`<workbook conformance="strict">`
	`…`
	`</workbook>`
	
	This document has a conformance attribute value of `strict`, therefore it conforms to the SML Strict conformance class. *end example*]
	
	The possible values for this attribute are defined by the ST_ConformanceClass simple type (§22.9.2.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Workbook) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.2.28 workbookPr (Workbook Properties)

This element defines a collection of workbook properties.

[*Example*:

```
<workbookPr showObjects="none" saveExternalLinkValues="0"
  defaultThemeVersion="123820"/>
```

end example]

Attributes	Description
allowRefreshQuery (Allow Refresh Query)	Specifies a boolean value that indicates whether the application will refresh query tables in this workbook. A value of `1` or `true` indicates the application will refresh query tables when the workbook is loaded. A value of `0` or `false` indicates the application will not refresh query tables. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoCompressPictures (Auto Compress Pictures)	Specifies a boolean value that indicates the application automatically compressed pictures in the workbook. A value of `1` or `true` indicates the application automatically compresses pictures of the workbook. When a picture is compressed, the application: • Reduces resolution (to 96 dots per inch (dpi) for Web and 200 dpi for print), and unnecessary information is discarded. • Discards extra information. [*Example*: When a picture has been cropped or resized, the "hidden" parts of the picture are stored in the file. *end example*] • Compress the picture, if possible. A value of `0` or `false` indicates the application does not compress pictures in this workbook. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
backupFile (Create Backup File)	Specifies a boolean value that indicates whether the application creates a backup of the workbook on save. A value of `1` or `true` indicates the application creates a backup of the workbook on save.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates the application does not create a backup. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
checkCompatibility (Check Compatibility On Save)	Specifies a boolean value that indicates whether the application checks for compatibility when saving this workbook to older file formats. A value of `1` or `true` indicates the application performs a compatibility check when saving to legacy binary formats. A value of `0` or `false` indicates the application does not perform a compatibility check when saving to legacy binary formats. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
codeName (Code Name)	Specifies the codename of the application that created this workbook. Use this attribute to track file content in incremental releases of the application. The possible values for this attribute are defined by the W3C XML Schema string datatype.
date1904 (Date 1904)	Value that indicates whether to use a 1900 or 1904 date system when converting serial date-times in the workbook to dates. A value of `1` or `true` indicates the workbook uses the 1904 date system. A value of `0` or `false` indicates the workbook uses the 1900 date system. (See §18.17.4.1 for the definition of the date systems.) The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultThemeVersion (Default Theme Version)	Specifies the default version of themes to apply in the workbook. The value for defaultThemeVersion depends on the application. SpreadsheetML defaults to the form [version][build], where [version] refers to the version of the application, and [build] refers to the build of the application when the themes in the user interface changed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
filterPrivacy (Filter Privacy)	Specifies a boolean value that indicates whether the application has inspected the workbook for personally identifying information (PII). If this flag is set, the application warns the user any time the user performs an action that will insert PII into the document. [*Example*: Inserting a comment might insert the user's name. *end example*] A value of `1` or `true` indicates the application will warn the user when they insert PII into the workbook. A value of `0` or `false` indicates the application will not warn the user when they insert PII into the workbook; the workbook has not been inspected for PII. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hidePivotFieldList (Hide Pivot Field List)	Specifies a boolean value that indicates whether a list of fields is shown for pivot tables in the application user interface. A value of `1` or `true` indicates a list of fields is show for pivot tables. A value of `0` or `false` indicates a list of fields is not shown for pivot tables. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
promptedSolutions (Prompted Solutions)	Specifies a boolean value that indicates whether the user has received an alert to load Smart Document components. A value of `1` or `true` indicates the user received an alert to load SmartDoc. A value of `0` or `false` indicates the user did not receive an alert. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
publishItems (Publish Items)	Specifies a boolean value that indicates whether the publish the workbook or workbook items to the application server. A value of `1` or `true` indicates that workbook items are published. A value of `0` or `false` indicates that the workbook is published.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The default value for this attribute is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
refreshAllConnections (Refresh all Connections on Open)	Specifies a boolean value that indicates whether the workbok shall refresh all the connections to data sources during load. The default value for this attribute is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saveExternalLinkValues (Save External Link Values)	Specifies a boolean value that indicates whether the application will cache values retrieved from other workbooks via an externally linking formula. Data is cached at save. A value of `1` or `true` indicates data from externally linked formulas is cached. A supporting part is written out containing a cached cell table from the external workbook. A value of `0` or `false` indicates data from externally linked formulas is not cached. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showBorderUnselectedTables (Show Border Unselected Table)	Specifies a boolean value that indicates whether a border is drawn around unselected tables in the workbook. A value of `1` or `true` indicates borders are drawn around unselected tables. A value of `0` or `false` indicates borders are not drawn around unselected tables. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showInkAnnotation (Show Ink Annotations)	Specifies a boolean value that indicates whether the book shows ink annotations. A value of `1` or `true` indicates that ink annotations are shown in the workbook. A value of `0` or `false` indicates that ink annotations are not shown in the workbook. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
showObjects (Show Objects)	Specifies how the application shows embedded objects in the workbook. This attribute it optional. The default value for this attribute is "all." The possible values for this attribute are defined by the ST_Objects simple type (§18.18.48).
showPivotChartFilter (Show Pivot Chart Filter)	Specifies a boolean value that indicates whether filtering options are shown for pivot charts in the workbook. A value of `1` or `true` indicates filtering options shall be shown for pivot charts. A value of `0` or `false` indicates filtering options shall not be shown. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
updateLinks (Update Links Behavior)	Specifies how the application updates external links when the workbook is opened. The default value for this attribute is userSet. The possible values for this attribute are defined by the ST_UpdateLinks simple type (§18.18.87).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WorkbookPr) is located in §A.2. *end note*]

18.2.29 workbookProtection (Workbook Protection)

This element specifies options for protecting data in the workbook. Applications might use workbook protection to prevent anyone from accidentally changing, moving, or deleting important data. This protection can be ignored by applications which choose not to support this optional protection mechanism.

When a password is to be hashed and stored in this element, it shall be hashed as defined below, starting from a UTF-16LE encoded string value. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

[*Note*: Worksheet or workbook element protection should not be confused with file security. It is not meant to make your workbook safe from unintentional modification, and cannot protect it from malicious modification. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
lockRevision (Lock Revisions)	Specifies a boolean value that indicates whether the workbook is locked for revisions. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lockStructure (Lock Structure)	Specifies a boolean value that indicates whether structure of workbook is locked. A value of `1` or `true` indicates the structure of the workbook is locked. Worksheets in the workbook can't be moved, deleted, hidden, unhidden, or renamed, and new worksheets can't be inserted. A value of `0` or `false` indicates the structure of the workbook is not locked. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lockWindows (Lock Windows)	Specifies a boolean value that indicates whether the windows that comprise the workbook are locked. A value of `1` or `true` indicates the workbook windows are locked. Windows are the same size and position each time the workbook is opened. A value of `0` or `false` indicates the workbook windows are not locked. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
revisionsAlgorithmName (Cryptographic Algorithm Name)	Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value for the revisionsHashValue attribute. The following values are reserved:

Value	Algorithm
`MD2`	Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]
`MD4`	Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used. [*Note*: It is recommended that applications should avoid using

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>

[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:

<pre>< … revisionsAlgorithmName="SHA-1"
revisionsHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>

The revisionsAlgorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm shall be used to generate a hash from the user-defined password. end example]

The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).</td></tr>
<tr><td>revisionsHashValue (Password Hash Value)</td><td>Specifies the hash value for the password stored for unlocking revisions in this workbook. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	XML element, and if the two values match, the protection shall no longer be enforced. If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook. [*Example*: Consider a SpreadsheetML document with the following information stored in one of its protection elements: `<… revisionsAlgorithmName="SHA-1" revisionsHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The revisionsHashValue attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the revisionsAlgorithmName attribute value of `SHA-1`) and that the resulting has value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
revisionsSaltValue (Salt Value for Password Verifier)	Specifies the salt that was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the revisionsHashValue attribute, and that shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… revisionsSaltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" revisionsHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The revisionsSaltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
revisionsSpinCount (Iterations to Run	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>Hashing Algorithm)</td><td>of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the revisionsHashValue attribute.

[Rationale: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). end rationale]

[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:

<code><… revisionsSpinCount="100000"
revisionHashValue="9oN7nWkCAyEZib1RomSJTjmPpcY=" /></code>

The revisionsSpinCount attribute value of <code>100000</code> specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the revisionsHashValue attribute. end example]

The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.</td></tr>
<tr><td>workbookAlgorithmName (Cryptographic Algorithm Name)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value for the workbookHashValue attribute.

The following values are reserved:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td><code>MD2</code></td><td>Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td><code>MD4</code></td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td><code>MD5</code></td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>

[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:

<pre>< … workbookAlgorithmName="SHA-1"
workbookHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The workbookAlgorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. end example]

The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).</td></tr>
<tr><td>workbookHashValue (Password Hash Value)</td><td>Specifies the hash value for the password stored for unlocking this workbook. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.

If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.

[Example: Consider a SpreadsheetML document with the following information stored in one of its protection elements:

<pre><… workbookAlgorithmName="SHA-1"</pre></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	workbookHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The workbookHashValue attribute value of 9oN7nWkCAyEZib1RomSJTjmPpCY= specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the workbookAlgorithmName attribute value of SHA-1) and that the resulting has value must be 9oN7nWkCAyEZib1RomSJTjmPpCY= for the protection to be disabled. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
workbookSaltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the workbookHashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: <… workbookSaltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" workbookHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /> The workbookSaltValue attribute value of ZUdHa+D8F/OAKP3I7ssUnQ== specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
workbookSpinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the workbookHashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… workbookSpinCount="100000" revisionHashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The workbookSpinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the workbookHashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WorkbookProtection) is located in §A.2. *end note*]

18.2.30 workbookView (Workbook View)

This element specifies a single Workbook view.

Units for window widths and other dimensions are expressed in twips. Twip measurements are portable between different display resolutions. The formula is (screen pixels) * (20 * 72) / (logical device dpi), where the logical device dpi can be different for x and y coordinates.

Attributes	Description
activeTab (Active Sheet Index)	Specifies an unsignedInt that contains the index to the active sheet in this book view. The default value for this attribute is 0. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
autoFilterDateGrouping (AutoFilter Date Grouping)	Specifies a boolean value that indicates whether to group dates when presenting the user with filtering options in the user interface. A value of `1` or `true` indicates that dates are grouped. A value of `0` or `false` indicates that dates are not grouped. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
firstSheet (First Sheet)	Specifies the index to the first sheet in this book view.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The default value for this attribute is 0. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
minimized (Minimized)	Specifies a boolean value that indicates whether the workbook window is minimized. A value of `1` or `true` indicates the workbook window is minimized. A value of `0` or `false` indicates the workbook window is not minimized. The default value for this attribute is false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showHorizontalScroll (Show Horizontal Scroll)	Specifies a boolean value that indicates whether to display the horizontal scroll bar in the user interface. A value of `1` or `true` indicates that the horizontal scrollbar shall be shown. A value of `0` or `false` indicates that the horizontal scrollbar shall not be shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showSheetTabs (Show Sheet Tabs)	Specifies a boolean value that indicates whether to display the sheet tabs in the user interface. A value of `1` or `true` indicates that sheet tabs shall be shown. A value of `0` or `false` indicates that sheet tabs shall not be shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showVerticalScroll (Show Vertical Scroll)	Specifies a boolean value that indicates whether to display the vertical scroll bar. A value of `1` or `true` indicates the vertical scrollbar shall be shown. A value of `0` or `false` indicates the vertical scrollbar shall not be shown. The default value for this attribute is true. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
tabRatio (Sheet Tab Ratio)	Specifies ratio between the workbook tabs bar and the horizontal scroll bar. The default value for this attribute is 600. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
visibility (Visibility)	Specifies visible state of the workbook window. The default value for this attribute is "visible." The possible values for this attribute are defined by the ST_Visibility simple type (§18.18.89).
windowHeight (Window Height)	Specifies the height of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
windowWidth (Window Width)	Specifies the width of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xWindow (Upper Left Corner (X Coordinate))	Specifies the X coordinate for the upper left corner of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema int datatype.
yWindow (Upper Left Corner (Y Coordinate))	Specifies the Y coordinate for the upper left corner of the workbook window. The unit of measurement for this value is twips. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BookView) is located in §A.2. *end note*]

18.3 Worksheets

Sheets are the central structures within a workbook, and are where the user does most of their spreadsheet work. The most common type of sheet is the worksheet, which is represented as a grid of cells. Worksheet cells can contain text, numbers, dates, and formulas. Cells can be formatted as well. Workbooks usually contain more than one sheet. To aid in the analysis of data and making informed decisions, spreadsheet applications often implement features and objects which help calculate, sort, filter, organize, and graphically display information.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Since these features are often connected very tightly with the spreadsheet grid, these are also included in the sheet definition on disk.

Other types of sheets include chart sheets and dialog sheets.

Note that sheet information is organized into three main sections:

- Top-level sheet properties (everything before sheetData)
- The cell table (sheetData)
- Supporting sheet features (everything after sheetData)

18.3.1 Worksheets

The following elements define a sheet and its contents:

18.3.1.1 anchor (Object Cell Anchor)

This element specifies the position of an embedded object or embedded control.

[*Example*: The following example demonstrates an embedded object whose top-left corner is at the top-left point of the cell in the first column and first row and whose bottom-right corner is offset horizontally into the cell at the fifth column and eleventh row.

```
<oleObject … >
  <objectPr … >
    <anchor sizeWithCells="true">
      <from>
        <col>0</col>
        <colOff>0</colOff>
        <row>0</row>
        <rowOff>0</rowOff>
      </from>
      <to>
        <col>4</col>
        <colOff>182880</colOff>
        <row>10</row>
        <rowOff>0</rowOff>
      </to>
    <anchor>
  </objectPr>
</oleObject>
```

end example]

Attributes	Description
moveWithCells	Specifies that the object moves with its underlying cells.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Move With Cells)	[*Example*: `<anchor moveWithCells="true" … >` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sizeWithCells (Size With Cells)	Specifies that the object resizes with its underlying cells. [*Example*: `<anchor sizeWithCells="true" … >` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
z-order (Z-Order)	Specifies the Z-order index of the object. Higher numbers indicate higher Z-order. Objects with higher Z-order are rendered on top of objects with lower Z-order when they intersect or overlap. [*Example*: `<anchor z-order="10 " … >` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectAnchor) is located in §A.2. *end note*]

18.3.1.2 autoFilter (AutoFilter Settings)

AutoFilter temporarily hides rows based on a filter criteria, which is applied column by column to a table of data in the worksheet. This collection expresses AutoFilter settings.

[*Example*: This example expresses a filter indicating to 'show only values greater than 0.5'. The filter is being applied to the range B3:E8, and the criteria is being applied to values in the column whose colId is 1 (zero based column numbering, from left to right). Therefore any rows must be hidden if the value in that particular column is less than or equal to 0.5.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<autoFilter ref="B3:E8">
  <filterColumn colId="1">
    <customFilters>
      <customFilter operator="greaterThan" val="0.5"/>
    </customFilters>
  </filterColumn>
</autoFilter>
```

end example]

Attributes	Description
ref (Cell or Range Reference)	Reference to the cell range to which the AutoFilter is applied. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AutoFilter) is located in §A.2. *end note*]

18.3.1.3 brk (Break)

A row or column break to use when paginating a worksheet. [*Note*: See §18.18.69 for more information on worksheet views. *end note*]

Attributes	Description
id (Id)	Zero-based row or column Id of the page break. Breaks occur above the specified row and left of the specified column. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
man (Manual Page Break)	Manual Break flag. 1 means the break is a manually inserted break. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
max (Maximum)	Zero-based index of end row or column of the break. For row breaks, specifies column index; for column breaks, specifies row index. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
min (Minimum)	Zero-based index of start row or column of the break. For row breaks, specifies column index; for column breaks, specifies row index. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
pt (Pivot-Created Page Break)	Flag indicating that a PivotTable created this break. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Break) is located in §A.2. *end note*]

18.3.1.4 c (Cell)

This collection represents a cell in the worksheet. Information about the cell's location (reference), value, data type, formatting, and formula is expressed here.

[*Example*: This example shows the information stored for a cell whose address in the grid is C6, whose style index is '6', and whose value metadata index is '15'. The cell contains a formula as well as a calculated result of that formula.

```
<c r="C6" s="1" vm="15">
  <f>CUBEVALUE("xlextdat9 Adventure Works",C$5,$A6)</f>
  <v>2838512.355</v>
</c>
```

end example]

While a cell can have a formula element f and a value element v, when the cell's type t is `inlineStr` then only the element is is allowed as a child element.

[*Example*:

Here is an example of expressing a string in the cell rather than using the shared string table.

```
<row r="1" spans="1:1">
  <c r="A1" t="inlineStr">
    <is><t>This is inline string example</t></is>
  </c>
</row>
```

end example]

Attributes	Description
cm (Cell Metadata Index)	The zero-based index of the cell metadata record associated with this cell. Metadata information is found in the Metadata Part. Cell metadata is extra information stored at the cell level, and is attached to the cell (travels through moves, copy / paste, clear, etc). Cell metadata is not accessible via formula reference.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ph (Show Phonetic)	A Boolean value indicating if the spreadsheet application should show phonetic information. Phonetic information is displayed in the same cell across the top of the cell and serves as a 'hint' which indicates how the text should be pronounced. This should only be used for East Asian languages. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Reference)	An A1 style reference to the location of this cell The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
s (Style Index)	The index of this cell's style. Style records are stored in the Styles Part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
t (Cell Data Type)	An enumeration representing the cell's data type. The possible values for this attribute are defined by the ST_CellType simple type (§18.18.11).
vm (Value Metadata Index)	The zero-based index of the value metadata record associated with this cell's value. Metadata records are stored in the Metadata Part. Value metadata is extra information stored at the cell level, but associated with the value rather than the cell itself. Value metadata is accessible via formula reference. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cell) is located in §A.2. *end note*]

18.3.1.5 cellSmartTag (Cell Smart Tag)

Single smart tag associated with a cell. There can be more than one cellSmartTag for a cell.

Attributes	Description
deleted (Deleted)	Boolean flag indicating that the application shouldn't display a particular smart tag in the cell. [*Example*: When the user has chosen to explicitly remove the Smart Tag by interacting with the application's user interface. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
type (Smart Tag Type Index)	Book-level zero-based index of the smart tag type. This index references a <smartTagType> element in the <smartTagTypes> collection in the workbook start part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xmlBased (XML Based)	Boolean flag indicating the Smart Tag recognition is triggered because the cell is associated with an XML map (schema-based semantic recognition), as contrasted with the more usual cell-content-based recognition of smart tags. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellSmartTag) is located in §A.2. *end note*]

18.3.1.6 cellSmartTagPr (Smart Tag Properties)

Represents a single property of a smart tag in a cell; contains a key-value pair.

Attributes	Description
key (Key Name)	Key name of a single property of a smart tag in a cell. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
val (Value)	String value of a single property of a smart tag in a cell. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellSmartTagPr) is located in §A.2. *end note*]

18.3.1.7 cellSmartTags (Cell Smart Tags)

The element is used to label the cell with a smart tag. A cell can be determined to have semantic meaning and the cell containing this data can be labeled with a smart tag. The actions which can be taken depend on the semantic meaning of the data and the actions that the application decides to associate with that type of smart tag.

[*Example*: If you recently sent mail to "Chad Rothschiller", and you type the name into a cell on the worksheet, the name is recognized and given a smart tag with actions you can take including Send Mail, Schedule a Meeting, Open Contact, or Add to Contacts.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

An application can decide that the smart tag indicators appear in the cell in the worksheet.

This collection represents a collection of smart tags on a cell.

[*Example*: This example expresses a smart tag associated with cell A1. The @type is used to associate this smart tag with a workbook-level smart tag type defined in the workbook start part.

```
<cellSmartTags r="A1">
  <cellSmartTag type="0"/>
</cellSmartTags>
```

end example]

Attributes	Description
r (Reference)	Reference to the cell that contains this set of smart tags. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellSmartTags) is located in §A.2. *end note*]

18.3.1.8 cellWatch (Cell Watch Item)

The watch window is a single UI location where the application user can keep track of certain cell formulas & values which they have chosen to be in the set of watched cells. This element expresses the cell address of a cell being watched. It is always a reference to a single cell.

Attributes	Description
r (Reference)	Cell reference of the cell being watched. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellWatch) is located in §A.2. *end note*]

18.3.1.9 cellWatches (Cell Watch Items)

Collection of cells on this worksheet being watched in the 'watch window'.

[*Example*: In this example, cells B3 and B4 are being watched.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<cellWatches>
  <cellWatch r="B3"/>
  <cellWatch r="B4"/>
</cellWatches>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellWatches) is located in §A.2. *end note*]

18.3.1.10 cfRule (Conditional Formatting Rule)

This collection represents a description of a conditional formatting rule.

[*Example*:

This example shows a conditional formatting rule highlighting cells whose values are greater than 0.5. Note that in this case the content of `<formula>` is a static value, but can also be a formula expression.

```
<conditionalFormatting sqref="E3:E9">
  <cfRule type="cellIs" dxfId="0" priority="1" operator="greaterThan">
    <formula>0.5</formula>
  </cfRule>
</conditionalFormatting>
```

end example]

Only rules with a type attribute value of `expression` support formula syntax.

Attributes	Description
aboveAverage (Above Or Below Average)	Indicates whether the rule is an "above average" rule. 1 indicates 'above average'. This attribute is ignored if type is not equal to `aboveAverage`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bottom (Bottom N)	Indicates whether a "top/bottom n" rule is a "bottom n" rule. 1 indicates 'bottom'. This attribute is ignored if type is not equal to `top10`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dxfId (Differential Formatting Id)	This is an index to a dxf element in the Styles Part indicating which cell formatting to apply when the conditional formatting rule criteria is met. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
equalAverage (Equal Average)	Flag indicating whether the 'aboveAverage' and 'belowAverage' criteria is inclusive of the average itself, or exclusive of that value. 1 indicates to include the average value in the criteria. This attribute is ignored if type is not equal to `aboveAverage`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
operator (Operator)	The operator in a "cell value is" conditional formatting rule. This attribute is ignored if type is not equal to `cellIs` The possible values for this attribute are defined by the ST_ConditionalFormattingOperator simple type (§18.18.15).
percent (Top 10 Percent)	Indicates whether a "top/bottom n" rule is a "top/bottom n percent" rule. This attribute is ignored if type is not equal to `top10`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
priority (Priority)	The priority of this conditional formatting rule. This value is used to determine which format should be evaluated and rendered. Lower numeric values are higher priority than higher numeric values, where 1 is the highest priority. The possible values for this attribute are defined by the W3C XML Schema int datatype.
rank (Rank)	The value of "n" in a "top/bottom n" conditional formatting rule. This attribute is ignored if type is not equal to `top10`. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
stdDev (StdDev)	The number of standard deviations to include above or below the average in the conditional formatting rule. This attribute is ignored if type is not equal to `aboveAverage`. If a value is present for stdDev and the rule type = `aboveAverage`, then this rule is automatically an "above or below N standard deviations" rule. The possible values for this attribute are defined by the W3C XML Schema int datatype.
stopIfTrue (Stop If True)	If this flag is 1, no rules with lower priority shall be applied over this rule, when this rule evaluates to true. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
text (Text)	The text value in a "text contains" conditional formatting rule. This attribute is ignored if type is not equal to `containsText`. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
timePeriod (Time Period)	The applicable time period in a "date occurring…" conditional formatting rule. This attribute is ignored if type is not equal to `timePeriod`. The possible values for this attribute are defined by the ST_TimePeriod simple type (§18.18.82).
type (Type)	Type of conditional formatting rule. The possible values for this attribute are defined by the ST_CfType simple type (§18.18.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CfRule) is located in §A.2. *end note*]

18.3.1.11 cfvo (Conditional Format Value Object)

Describes the values of the interpolation points in a gradient scale.

[*Example*: This example demonstrates a color scale conditional formatting rule, which defines a color for the minimum value in the range of cell values, a color for the midpoint value, and a color for the maximum value in the in the range of cell values. Information is given about how to define the midpoint. In this case, it is the 50 percent mark.

```
<colorScale>
  <cfvo type="min" val="0"/>
  <cfvo type="percent" val="50"/>
  <cfvo type="max" val="0"/>
  <color rgb="FFFF0000"/>
  <color rgb="FFFFFF00"/>
  <color rgb="FF00B050"/>
</colorScale>
```

The first `<cfvo>` element corresponds with the first `<color>` definition, and so on.

end example]

Attributes	Description
gte (Greater Than Or Equal)	For icon sets, determines whether this threshold value uses the greater than or equal to operator. 0 indicates 'greater than' is used instead of 'greater than or equal to'. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
type (Type)	The type of this conditional formatting value object. [*Example*: 'min' and 'max' would be used (in conjunction with `@val`) to express the lower and upper values to be used in a gradient. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_CfvoType simple type (§18.18.13).
val (Value)	The value of this conditional formatting value object. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cfvo) is located in §A.2. *end note*]

18.3.1.12 chartsheet (Chart Sheet)

This is the root element of Chartsheet Parts in a SpreadsheetML document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Chartsheet) is located in §A.2. *end note*]

18.3.1.13 col (Column Width & Formatting)

Defines column width and column formatting for one or more columns of the worksheet.

[*Example*: This example shows that column 5 (E) has width and style information applied.

```
<col min="5" max="5" width="9.140625" style="3"/>
```

end example]

Attributes	Description
bestFit (Best Fit Column Width)	Flag indicating if the specified column(s) is set to 'best fit'. 'Best fit' is set to `true` under these conditions: • The column width has never been manually set by the user, AND • The column width is not the default width • • 'Best fit' means that when numbers are typed into a cell contained in a 'best fit' column, the column width should automatically resize to display the number. [*Note*: In best fit cases, column width must not be made smaller, only larger. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
collapsed (Collapsed)	Flag indicating if the outlining of the affected column(s) is in the collapsed state. See description of row collapsed and outlinePr element's summaryBelow and summaryRight attributes for detailed information. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
customWidth (Custom Width)	Flag indicating that the column width for the affected column(s) is different from the default or has been manually set. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hidden (Hidden Columns)	Flag indicating if the affected column(s) are hidden on this worksheet. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
max (Maximum Column)	Last column affected by this 'column info' record. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
min (Minimum Column)	First column affected by this 'column info' record. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
outlineLevel (Outline Level)	Outline level of affected column(s). Range is 0 to 7. See description of outlinePr element's summaryBelow and summaryRight attributes for detailed information. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
phonetic (Show Phonetic Information)	Flag indicating if the phonetic information should be displayed by default for the affected column(s) of the worksheet. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
style (Style)	Default style for the affected column(s). Affects cells not yet allocated in the column(s). In other words, this style applies to new columns. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
width (Column Width)	Column width measured as the number of characters of the maximum digit width of the numbers 0, 1, 2, …, 9 as rendered in the normal style's font. There are 4 pixels of margin padding (two on each side), plus 1 pixel padding for the gridlines. width = Truncate([{Number of Characters} * {Maximum Digit Width} + {5 pixel padding}]/{Maximum Digit Width}*256)/256 [*Example*: Using the Calibri font as an example, the maximum digit width of 11 point font size is 7 pixels (at 96 dpi). In fact, each digit is the same width for this font. Therefore, if the cell width is 8 characters wide, the value of this attribute must be Truncate([8*7+5]/7*256)/256 = 8.7109375. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	To translate the value of width in the file into the column width value at runtime (expressed in terms of pixels), use this calculation: =Truncate(((256 * {width} + Truncate(128/{Maximum Digit Width}))/256)*{Maximum Digit Width}) [*Example*: Using the same example as above, the calculation would be Truncate(((256*8.7109375+Truncate(128/7))/256)*7) = 61 pixels. *end example*] To translate from pixels to character width, use this calculation: =Truncate(({pixels}-5)/{Maximum Digit Width} * 100+0.5)/100 [*Example*: Using the example above, the calculation would be Truncate((61-5)/7*100+0.5)/100 = 8 characters. *end example*] [*Note*: when wide borders are applied, part of the left/right border must overlap with the 2 pixel padding on each side. Wide borders do not affect the width calculation of the column. *end note*] [*Note*: When the sheet is in the mode to view formulas instead of values, the pixel width of the column is doubled. *end note*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Col) is located in §A.2. *end note*]

18.3.1.14 colBreaks (Vertical Page Breaks)

A collection of column breaks (§18.3.1.3).

[*Example*:

In this example, a page break has been inserted at C3 (the break occurs left and above C3).

```
<colBreaks count="1" manualBreakCount="1">
  <brk id="2" max="1048575" man="1"/>
</colBreaks>
```

end example]

Attributes	Description
count (Page Break	Number of breaks in the collection.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Count)	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
manualBreakCount (Manual Break Count)	Number of manual breaks in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageBreak) is located in §A.2. *end note*]

18.3.1.15 color (Data Bar Color)

One of the colors associated with the data bar or color scale.

The auto attribute shall not be used in the context of data bars.

Attributes	Description
auto (Automatic)	A boolean value indicating the color is automatic and system color dependent. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indexed (Index)	Indexed color value. Only used for backwards compatibility. References a color in indexedColors. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rgb (Alpha Red Green Blue Color Value)	Standard Alpha Red Green Blue color value (ARGB). The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
theme (Theme Color)	A zero-based index into the `<clrScheme>` collection (§20.1.6.2), referencing a particular `<sysClr>` or `<srgbClr>` value expressed in the Theme part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
tint (Tint)	Specifies the tint value applied to the color. If tint is supplied, then it is applied to the RGB value of the color to determine the final color applied. The tint value is stored as a double from -1.0 .. 1.0, where -1.0 means 100% darken and 1.0 means 100% lighten. Also, 0.0 means no change.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	In loading the RGB value, it is converted to HLS where HLS values are (0..HLSMAX), where HLSMAX is currently 255. [*Example*: Here are some examples of how to apply tint to color: **If (tint < 0)** Lum' = Lum * (1.0 + tint) For example: Lum = 200; tint = -0.5; Darken 50% Lum' = 200 * (0.5) => 100 For example: Lum = 200; tint = -1.0; Darken 100% (make black) Lum' = 200 * (1.0-1.0) => 0 **If (tint > 0)** Lum' = Lum * (1.0-tint) + (HLSMAX – HLSMAX * (1.0-tint)) For example: Lum = 100; tint = 0.75; Lighten 75% Lum' = 100 * (1-.75) + (HLSMAX – HLSMAX*(1-.75)) = 100 * .25 + (255 – 255 * .25) = 25 + (255 – 63) = 25 + 192 = 217 For example: Lum = 100; tint = 1.0; Lighten 100% (make white) Lum' = 100 * (1-1) + (HLSMAX – HLSMAX*(1-1)) = 100 * 0 + (255 – 255 * 0) = 0 + (255 – 0) = 255 *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.2. *end note*]

18.3.1.16 colorScale (Color Scale)

Describes a gradated color scale in this conditional formatting rule.

[*Example*:

©ISO/IEC 2016 – All rights reserved

```
<colorScale>
  <cfvo type="min" val="0"/>
  <cfvo type="max" val="0"/>
  <color theme="5"/>
  <color rgb="FFFFEF9C"/>
</colorScale>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorScale) is located in §A.2. *end note*]

18.3.1.17 cols (Column Information)

Information about whole columns of the worksheet.

[*Example*:

This example shows that column 4 (D) has 'best fit' applied to it, which is also a custom width. Also, column 5 (E) is listed as having a custom width and a style applied at the column level (as opposed to the cell level).

```
<cols>
  <col min="4" max="4" width="12" bestFit="1" customWidth="1"/>
  <col min="5" max="5" width="9.140625" style="3"/>
</cols>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cols) is located in §A.2. *end note*]

18.3.1.18 conditionalFormatting (Conditional Formatting)

A Conditional Format is a format, such as cell shading or font color, that a spreadsheet application can automatically apply to cells if a specified condition is true. This collection expresses conditional formatting rules applied to a particular cell or range.

[*Example*: This example applies a 'top10' rule to the cells C3:C8. The `@dxfId` references the formatting (defined in the styles part) to be applied to cells that match the criteria.

```
<conditionalFormatting sqref="C3:C8">
  <cfRule type="top10" dxfId="1" priority="3" rank="2"/>
</conditionalFormatting>
```

end example]

Attributes	Description
pivot (PivotTable Conditional	Flag indicating if this is conditional formatting associated with a PivotTable.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Formatting)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sqref (Sequence of Refernces)	Range over which these conditional formatting rules apply. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConditionalFormatting) is located in §A.2. *end note*]

18.3.1.19 control (Embedded Control)

A single embedded control.

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This relationship ID references an Embedded Control Data part which contains control-specific properties and state information about this particular embedded control. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Control Name)	The code name of the control. The possible values for this attribute are defined by the W3C XML Schema string datatype.
shapeId (Shape Id)	ID of the drawing shape in the DrawingML part with which this control is associated. The drawing is used to draw the control in the sheet. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Control) is located in §A.2. *end note*]

18.3.1.20 controlPr (Embedded Control Properties)

This element specifies the visual, positional and cell linkage properties of an embedded control.

[*Example*: The following example demonstrates an non-printing embedded control linked to cell A4 that is represented as an image:

```
<controls>
  <control … >
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <controlPr print="false" autoLine="false" linkedCell="$A$4" cf="pict"
  r:id="rId5">
        <anchor sizeWithCells="true">
          <from> … </from>
          <to> … </to>
        </anchor>
      </controlPr>
    </control>
  </controls>
```

end example]

Attributes	Description
altText (Alternative Text)	Specifies alternative text for the object, for use by assistive technologies or applications. [*Example*: `<controlPr altText="Alternate text"… />` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
autoFill (Automatic Fill Flag)	Specifies whether the object's fill formatting is provided automatically by the application. [*Example*: `<controlPr autoFill="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoLine (Automatic Line Flag)	Specifies whether the object's line formatting is provided automatically by the application. [*Example*: `<controlPr autoLine="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoPict (Automatic	Specifies whether the object's size is formatted automatically by the application.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Size Flag)	[*Example*: `<controlPr autoPict="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
cf (Image Format)	Specifies the image format used to render the object. [*Example*: `<controlPr cf="pict" … />` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
defaultSize (Default Size Flag)	Specifies whether the object is at its default size. [*Example*: `<controlPr defaultSize="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
disabled (Disabled Flag)	Specifies whether the object is allowed to run an attached macro. [*Example*: `<controlPr disabled="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Relationship ID for Embedded Control Properties) Namespace: http://purl.oclc.or	Specifies the relationship ID for the relationship which contains the properties for this embedded control. This property bag is contained in a separate part within the package. The relationship explicitly targeted by this attribute shall be of relationship type http://purl.oclc.org/ooxml/officeDocument/relationships/control or the document shall be conisdered non-conformant.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
g/ooxml/officeDocument/relationships	If this attribute is omitted, then the embedded control shall be given no property bag when instantiated. [*Example*: Consider the following WordprocessingML markup for an embedded control in a document: `<w:control r:id="rId5" w:id="CheckBox1" w:name="CheckBox1" w:shapeid="_x0000_s1027" w:class="shape" w:w="145" w:h="28" w:align="left" />` The id attribute in the relationship reference namespace specifies that the relationship with relationship ID `rId5` must contain the property data for this embedded control. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
linkedCell (Linked Formula)	Specifies the cell the control is linked to, using standard cell A1-style reference syntax as described in §18.17.2.3.1. The value in the linked cell and the index of the selected item in the object are linked together. This link is ignored if the control allows multiple selections. [*Example*: `<controlPr linkedCell="$A$4" … />` *end example*] The possible values for this attribute are defined by the ST_Formula simple type (§18.18.35).
listFillRange (List Items Source Range)	Specifies the range of source data cells used to populate the list box, using standard A1-style cell reference syntax as described in §18.17.2.3.1. [*Example*: `<controlPr listFillRange="$A$1:$A$15" … />` *end example*] The possible values for this attribute are defined by the ST_Formula simple type (§18.18.35).
locked (Locked Flag)	Specifies that the object is locked when the sheet is protected. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<controlPr locked="false" … /> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Custom Function)	Specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] [*Example*: <controlPr macro="Button1_Click()" … /> *end example*] The possible values for this attribute are defined by the ST_Formula simple type (§18.18.35).
print (Print Flag)	Specifies whether the object is printed when the document is printed. [*Example*: <controlPr print="false" … /> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
recalcAlways (Recalculation Flag)	Specifies whether the object is always included in recalculation. This is used by controls that reference cells in the spreadsheet to update themselves when the spreadsheet changes. [*Example*: <controlPr recalcAlways="true" … /> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uiObject (UI Object Flag)	Specifies whether the object is a UI-only object. Applications should prevent UI-only objects from being selected and edited in their user interface. [*Example*: <objectPr uiObject="true" … />

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ControlPr) is located in §A.2. *end note*]

18.3.1.21 controls (Embedded Controls)

Worksheets can have embedded controls embedded in them. This collection is a listing of embedded controls in this worksheet. This collection is used to reference individual Embedded Control Data part definitions, enumerate the code name of each control, and reference drawing information used to draw the control.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Controls) is located in §A.2. *end note*]

18.3.1.22 customPr (Custom Property)

The custom property element provides a mechanism to store name/value pairs of arbitrary user-defined data. The name is stored in the attribute name, the arbitrary data is stored in the binary part referenced by the relationshipId.

[*Note*: There is nothing in the binary part except the arbitrary data itself.

Custom XML Data Properties provide a preferred mechanism for storing arbitrary data. The customPr supports legacy third-party document components, as well as those situations that have a stringent need for binary parts. *end note*]

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This relationship references the binary part containing the specified custom properties. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Custom Property Name)	Name of the custom property The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomProperty) is located in §A.2. *end note*]

18.3.1.23 customProperties (Custom Properties)

This collection is used to reference binary parts containing arbitrary user-defined data.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomProperties) is located in §A.2. *end note*]

18.3.1.24 customSheetView (Custom Chart Sheet View)

This element defines custom view properties for chart sheets. [*Note*: See customSheetView (§18.3.1.25) for an example. *end note*]

Attributes	Description
guid (GUID)	Unique identifier of this custom view The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
scale (Print Scale)	Print scaling, representing percent values. The values of this attribute shall be restricted to the range from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
state (Visible State)	Visibility state of the sheet. The possible values for this attribute are defined by the ST_SheetState simple type (§18.18.68).
zoomToFit (Zoom To Fit)	Flag indicating whether chart sheet is zoom to fit window. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomChartsheetView) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.3.1.25 customSheetView (Custom Sheet View)

This collection stores information pertaining to one custom sheet view definition. A custom view is a collection of settings defining a particular view of the sheet. These views can be selected by the user for quick access to predefined views of the sheet.

[*Example*: This example indicates that there is both a horizontal and vertical split in the sheet view, and that the top left cell of the bottom right pane is F7. Page margin, print options, page setup, and header / footer information is also stored with this view.

```
<customSheetView guid="{F3A061A9-D5FD-4F9C-A7CD-483AD476BA25}"
  sizeWithWindow="0">
  <pane xSplit="5" ySplit="6" topLeftCell="F7"/>
  <selection/>
  <pageMargins left="0.7" right="0.7" top="0.75" bottom="0.75" header="0.3"
    footer="0.3"/>
  <printOptions gridLinesSet="0"/>
  <pageSetup paperSize="0" scale="0" orientation="portrait" printDriver="0"
    horizontalDpi="0" verticalDpi="0" copies="0"/>
  <headerFooter/>
</customSheetView>
```

end example]

Attributes	Description
colorId (Color Id)	Index to the color value for the text in row/column headings and gridlines for this custom view. This is an 'index color value' (ICV) rather than rgb value. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
filter (Filtered List)	Flag indicating whether the view contains a filtered range. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
filterUnique (Filter)	Indicates whether an advanced filter has been applied, and the option to filter out duplicate records from the data list has been selected, in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fitToPage (Fit To Page)	Flag indicating whether this view should be fit to page when printing this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
guid (GUID)	Unique identifier of this custom view. This is used to ensure uniqueness. It is generated when the view is created. Shall correspond to a customWorkbookView guid value in the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	workbook Start Part. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
hiddenColumns (Hidden Columns)	Flag indicating that there is one or more hidden column(s) in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hiddenRows (Hidden Rows)	Flag indicating that there is one or more hidden row(s) in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
outlineSymbols (Show Outline Symbols)	Flag indicating whether outline symbols are displayed in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
printArea (Print Area Defined)	Flag indicating whether a print area is defined as part of this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
scale (Print Scale)	Print scaling for this custom view. The values of this attribute shall be restricted to the range from 10 to 400. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
showAutoFilter (Show AutoFilter Drop Down Controls)	Flag indicating whether the autofilter dropdown buttons are visible in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showFormulas (Show Formulas)	Flag indicating whether formulas are shown in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showGridLines	Flag indicating whether gridlines are shown in this custom view.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Show Grid Lines)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showPageBreaks (Show Page Breaks)	Flag indicating whether page breaks are shown in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRowCol (Show Headers)	Flag indicating whether row and column headers are shown in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRuler (Show Ruler)	Flag indicating whether to show the ruler in this custom view. Only applicable if this Custom View is in page layout view (§18.18.69). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
state (Visible State)	Visibility state for this custom view. The possible values for this attribute are defined by the ST_SheetState simple type (§18.18.68).
topLeftCell (Top Left Visible Cell)	Location of the top left visible cell in the bottom right pane in this custom view (when in Left-to-Right mode). The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
view (View Type)	Indicates the view type for this Custom View The possible values for this attribute are defined by the ST_SheetViewType simple type (§18.18.69).
zeroValues (Show Zero Values)	Flag indicating whether the window should display 0 (zero) values in this custom view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomSheetView) is located in §A.2. *end note*]

18.3.1.26 customSheetViews (Custom Chart Sheet Views)

Collection of custom Chart Sheet View information.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomChartsheetViews) is located in §A.2. *end note*]

©ISO/IEC 2016 – All rights reserved

18.3.1.27 customSheetViews (Custom Sheet Views)

This is a collection of custom sheet views.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomSheetViews) is located in §A.2. *end note*]

18.3.1.28 dataBar (Data Bar)

Describes a data bar conditional formatting rule.

[*Example*:

In this example a data bar conditional format is expressed, which spreads across all cell values in the cell range, and whose color is blue.

```
<dataBar>
  <cfvo type="min" val="0"/>
  <cfvo type="max" val="0"/>
  <color rgb="FF638EC6"/>
</dataBar>
```

end example]

The length of the data bar for any cell can be calculated as follows:

Data bar length = minLength + (cell value - minimum value in the range) / (maximum value in the range - minimum value in the range) * (maxLength - minLength),

where min and max length are a fixed percentage of the column width (by default, 10% and 90% respectively.)

The minimum difference in length (or increment amount) is 1 pixel.

Attributes	Description
maxLength (Maximum Length)	The maximum length of the data bar, as a percentage of the cell width. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
minLength (Minimum Length)	The minimum length of the data bar, as a percentage of the cell width. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
showValue (Show Values)	Indicates whether to show the values of the cells on which this data bar is applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataBar) is located in §A.2. *end note*]

18.3.1.29 dataConsolidate (Data Consolidate)

Data consolidation settings. The dataRefs are the set of source ranges containing data to consolidate. The function indicates the function that shall be used to consolidate the data.

[*Example*:

This example demonstrates consolidating the ranges A1:C1 and A3:C3 by using the 'count' function.

```
<dataConsolidate function="count">
  <dataRefs count="2">
    <dataRef ref="A1:C1" sheet="Sheet1"/>
    <dataRef ref="A3:C3" sheet="Sheet1"/>
  </dataRefs>
</dataConsolidate>
```

end example]

Attributes	Description
function (Function Index)	Indicates which function to use when consolidating the ranges. The possible values for this attribute are defined by the ST_DataConsolidateFunction simple type (§18.18.17).
link (Link)	Create links to source data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
startLabels (Use Starting Column Labels)	Use labels in first column. Both startLabels and topLabels can be `true` at the same time. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
topLabels (Labels In Top Row)	Use labels in top row. Both leftLabels and topLabels can be `true` at the same time. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataConsolidate) is located in §A.2. *end note*]

18.3.1.30 dataRef (Data Consolidation Reference)

A single data consolidate reference. One dataRef shall use either name or sheet & ref, but not both on the same dataRef.

Attributes	Description
id (relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Used only when the source range is external to this workbook. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Named Range)	Named range, either in this workbook or the external workbook referenced by r:Id. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ref (Reference)	Cell range. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sheet (Sheet Name)	Sheet name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataRef) is located in §A.2. *end note*]

18.3.1.31 dataRefs (Data Consolidation References)

Data consolidate reference collection.

Attributes	Description
count (Data Consolidation Reference Count)	Count of data consolidate references. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataRefs) is located in §A.2. *end note*]

18.3.1.32 dataValidation (Data Validation)

A single item of data validation defined on a range of the worksheet.

Attributes	Description
allowBlank (Allow Blank)	A boolean value indicating whether the data validation allows the use of empty or blank entries. 1 means empty entries are OK and do not violate the validation constraints. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
error (Error Message)	Message text of error alert. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
errorStyle (Data Validation Error Style)	The style of error alert used for this data validation. The possible values for this attribute are defined by the ST_DataValidationErrorStyle simple type (§18.18.18).
errorTitle (Error Alert Text)	Title bar text of error alert. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
imeMode (IME Mode Enforced)	The IME (input method editor) mode enforced by this data validation. Only applies for these languages: • Chinese Simplified • Chinese Traditional • Japanese • Korean • • When imeMode is set, the input for the cell can be restricted to specific sets of characters, as specified by the value of imeMode. See the simple type referenced below for additional details. • When imeMode is set but the application's language is not one of the languages listed above, then the default value is `noControl`. The possible values for this attribute are defined by the ST_DataValidationImeMode simple type (§18.18.19).
operator (Operator)	The relational operator used with this data validation. The possible values for this attribute are defined by the ST_DataValidationOperator simple type (§18.18.20).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
prompt (Input Prompt)	Message text of input prompt. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
promptTitle (Prompt Title)	Title bar text of input prompt. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
showDropDown (Show Drop Down)	A boolean value indicating whether to display a dropdown combo box for a list type data validation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showErrorMessage (Show Error Message)	A boolean value indicating whether to display the error alert message when an invalid value has been entered, according to the criteria specified. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showInputMessage (Show Input Message)	A boolean value indicating whether to display the input prompt message. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sqref (Sequence of References)	Range over which data validation is applied. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).
type (Data Validation Type)	The type of data validation. The possible values for this attribute are defined by the ST_DataValidationType simple type (§18.18.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataValidation) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.3.1.33 dataValidations (Data Validations)

This collection expresses all data validation information for cells in a sheet which have data validation features applied.

Data validation is used to specify constraints on the data that can be entered into a cell. Additional UI can be provided to help the user select values (e.g., a dropdown control on the cell or hover text when the cell is active), and to help the user understand why a particular entry was disallowed (e.g., alerts and messages).

 ©ISO/IEC 2016 – All rights reserved

Various data types can be selected, and logical operators (e.g., greater than, less than, equal to, etc) can be used. Additionally, instead of specifying an explicit set of values that are permitted, a cell or range reference can be used.

An input message can be specified to help the user know what kind of value is expected, and a warning message (and warning type) can be specified to alert the user when they've entered data which is not permitted based on the data validations specified in the worksheet.

[*Example*:

```
<dataValidations count="1">
  <dataValidation type="whole" errorStyle="warning" operator="greaterThan"
    showInputMessage="1" showErrorMessage="1" errorTitle="Invalid Data"
    error="The value must be a whole number greater than 0."
    promptTitle="Whole Number"
    prompt="Please enter a whole number greater than 0." sqref="A1">
    <formula1>0</formula1>
  </dataValidation>
</dataValidations>
```

end example]

Attributes	Description
count (Data Validation Item Count)	The expected number of data validation items for this worksheet. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
disablePrompts (Disable Prompts)	A boolean value indicating whether all input prompts for the worksheet are disabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xWindow (Top Left Corner (X Coodrinate))	The x-coordinate (relative to window) of top-left corner of the data validation input prompt (textbox). This is per sheet, not per cell. Units in pixels. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
yWindow (Top Left Corner (Y Coordinate))	The y-coordinate (relative to window) of top-left corner of the data validation input prompt (textbox). This is per sheet, not per cell. Units in pixels. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataValidations) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.3.1.34 dialogsheet (Dialog Sheet)

This is the root element for Dialogsheet parts within a SpreadsheetML document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dialogsheet) is located in §A.2. *end note*]

18.3.1.35 dimension (Worksheet Dimensions)

This element specifies the used range of the worksheet. It specifies the row and column bounds of used cells in the worksheet. This is optional and is not required. Used cells include cells with formulas, text content, and cell formatting. When an entire column is formatted, only the first cell in that column is considered used.

[*Example*:

```
<dimension ref="A1:C2"/>
```

end example]

Attributes	Description
ref (Reference)	The row and column bounds of all cells in this worksheet. Corresponds to the range that would contain all c elements written under sheetData. Does not support whole column or whole row reference notation. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetDimension) is located in §A.2. *end note*]

18.3.1.36 drawing (Drawing)

This element indicates that the sheet contains drawing components built on the drawingML platform. The relationship Id references the part containing the drawingML definitions.

Attributes	Description
id (Relationship id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id referencing a part containing drawingML definitions for this worksheet. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.3.1.37 drawingHF (Drawing Reference in Header Footer)

This element specifies the usage of drawing objects to be rendered in the headers and footers of the sheet. It specifies an explicit relationship to the part containing the DrawingML shapes used in the headers and footers. It also indicates where in the headers and footers each shape belongs. One drawing object can appear in each of the left section, center section and right section of a header and a footer.

[*Example*: This example shows a worksheet with graphics in the header. The DrawingML part referred to by rId2 contains at least two objects. The object with ID 6 is shown in the left section of the header on the first page only. The object with ID 7 is shown in the left section of the header for the other pages.

```
<worksheet … >
  …
  <headerFooter differentFirst="1" … >
    …
  </headerFooter>
  <drawingHF r:id="rId2" lho="7" lhf="6"/>
</worksheet>
```

end example]

Attributes	Description
cfe (Center Footer for Even Pages)	Specifies the DrawingML shape to be used for the center section of the footer on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … cfe="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
cff (Center Footer for First Page)	Specifies the DrawingML shape to be used for the center section of the footer on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … cff="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
cfo (Center Footer for Odd Pages)	Specifies the DrawingML shape to be used for the center section of the footer on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the center section of the footer on both odd and even pages. [*Example*: `<drawingHF … cfo="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
che (Center Header for Even Pages)	Specifies the DrawingML shape to be used for the center section of the header on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … che="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
chf (Center Header for First Page)	Specifies the DrawingML shape to be used for the center section of the header on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … chf="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
cho (Center Header for Odd Pages)	Specifies the DrawingML shape to be used for the center section of the header on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the center section of the header on both odd and even pages. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<drawingHF … cho="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Relationship ID for Embedded Control Properties) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship to the DrawingML part that contains the drawing objects used in the header and footer. This DrawingML part is a separate part within the package. [*Example*: <drawingHF r:id="rId2" lho="7" lhf="6"/> The id attribute in the relationship reference namespace specifies that the relationship with relationship ID rId5 must contain the drawing objects used in the header and footer. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
lfe (Left Footer for Even Pages)	Specifies the DrawingML shape to be used for the left section of the footer on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: <drawingHF … lfe="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
lff (Left Footer for First Page)	Specifies the DrawingML shape to be used for the left section of the footer on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: <drawingHF … lff="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
lfo (Left Footer for Odd Pages)	Specifies the DrawingML shape to be used for the left section of the footer on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the left section of the footer on both odd and even pages. [*Example*: `<drawingHF … lfo="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
lhe (Left Header for Even Pages)	Specifies the DrawingML shape to be used for the left section of the header on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … lhe="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
lhf (Left Header for First Page)	Specifies the DrawingML shape to be used for the left section of the header on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … lhf="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
lho (Left Header for Odd Pages)	Specifies the DrawingML shape to be used for the left section of the header on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the left section of the header on both odd and even pages. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<drawingHF … lho="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rfe (Right Footer for Even Pages)	Specifies the DrawingML shape to be used for the right section of the footer on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: <drawingHF … rfe="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rff (Right Footer for First Page)	Specifies the DrawingML shape to be used for the right section of the footer on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: <drawingHF … rff="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rfo (Right Footer for Odd Pages)	Specifies the DrawingML shape to be used for the right section of the footer on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the right section of the footer on both odd and even pages. [*Example*: <drawingHF … rfo="5"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
rhe (Right Header for Even Pages)	Specifies the DrawingML shape to be used for the right section side of the header on even pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … rhe="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rhf (Right Header for First Page)	Specifies the DrawingML shape to be used for the right section of the header on the first page if the differentFirst attribute of the corresponding headerFooter element (§18.3.1.46) is true. [*Example*: `<drawingHF … rhf="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rho (Right Header for Odd Pages)	Specifies the DrawingML shape to be used for the right section of the header on odd pages if the differentOddEven attribute of the corresponding headerFooter element (§18.3.1.46) is true. If the differentOddEven attribute is false, this attribute specifies the DrawingML shape to be used for the right section of the header on both odd and even pages. [*Example*: `<drawingHF … rho="5"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DrawingHF) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.3.1.38 evenFooter (Even Page Footer)

Specifies the contents of the page footer used on even-numbered pages. [*Note*: Even page(s) in the sheet cannot be printed if the print area is specified to be a range such that it falls outside an even page's scope. *end note*]

If no even footer is specified, then the odd footer's value is assumed for even page footers.

The format of a footer is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.39 evenHeader (Even Page Header)

Specifies the contents of the page header used on even-numbered pages. [*Note*: Even page(s) in the sheet cannot be printed if the print area is specified to be a range such that it falls outside an even page's scope. *end note*]

If no even header is specified, then the odd header's value is assumed for even page headers.

The format of a header is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.40 f (Formula)

Formula for the cell. The formula expression is contained in the character node of this element.

[*Example*:

```
<f>SUM(C4:E4)</f>
```

end example]

The possible values for the t attribute are defined by the simple type ST_CellFormulaType, and are as follows:

Value	Description
array (Array formula)	Array formula. An array formula is a single formula, applied across a range of one or more cells. An array formula can return multiple results from one single calculation, the results spanning the cells in which it is contained (§18.17.2.7).
dataTable (Table formula)	Data table. A *data table* is a range of cells that shows how changing certain values in one or more formulas affects the results of those formulas. A data table provides a shortcut for calculating multiple versions in one operation, and a way to include the results of all of the different variations in a sheet. Both one- and two-input variable data tables can be created (see attribute dt2D).[*Example*: A one-input variable data table might be used to calculate how different interest rates affect a monthly mortgage payment, while a two-input variable data table might be used to calculate how different interest rates and loan terms will affect the mortgage payment. *end example*] In a one-input variable data table, values are listed either down a column (column-oriented) or across a row (row-oriented) (see attribute dtr). Formulas that are used in a one-input variable data table shall refer to an input cell (see attribute r1), the cell in which each input value from a data table is substituted. Any cell on a worksheet can be the input cell. Although the input cell does not need to be part of the data table, the formulas in data tables shall refer to that input cell. Two-input variable data tables use only one formula with two lists of input values. The formula shall refer to two input cells (see attributes r1 and r2). The top-left cell in the data table is called the *master cell*. [*Guidance*: It is recommended that Spreadsheet applications recalculate data tables whenever a worksheet is recalculated. *end guidance*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Value	Description
normal (Normal formula)	Normal cell formula (§18.17).
shared (Shared formula)	Shared formula. If a cell contains the same formula as another cell, the “shared” value can be used for the t attribute and the si attribute can be used to refer to the cell containing the formula. Two formulas are considered to be the same when their respective representations in R1C1-reference notation, are the same.

Attributes	Description
aca (Always Calculate Array)	Only applies to array formulas. `true` indicates that the entire array shall be calculated in full. If `false` the individual cells of the array shall be calculated as needed. The `aca` value shall be ignored unless the value of the corresponding t attribute is array. [*Note*: The primary case where an array formula must be calculated in part instead of in full is when some cells in the array depend on other cells that are semi-calculated, e.g., contains the function =RAND(). *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bx (Assigns Value to Name)	Specifies that this formula assigns a value to a name. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ca (Calculate Cell)	Indicates that this formula needs to be recalculated the next time calculation is performed. [*Example*: This is always set on volatile functions, like =RAND(), and circular references. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
del1 (Input 1 Deleted)	Whether the first input cell for data table has been deleted. Applies to data table formula only. Written on master cell of data table formula only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
del2 (Input 2 Deleted)	Whether the second input cell for data table has been deleted. Applies to data table formula only. Written on master cell of data table formula only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dt2D (Data Table 2-D)	Data table is two-dimentional. Only applies to the data tables function. Written on master cell of data table formula only. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
dtr (Data Table Row)	`true` if one-dimentional data table is a row, otherwise it's a column. Only applies to the data tables function. Written on master cell of data table formula only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r1 (Data Table Cell 1)	First input cell for data table. Only applies to the data tables array function "TABLE()". Written on master cell of data table formula only. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
r2 (Input Cell 2)	Second input cell for data table when dt2D is 1. Only applies to the data tables array function "TABLE()".Written on master cell of data table formula only. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
ref (Range of Cells)	Range of cells which the formula applies to. Only required for shared formula, array formula or data table. Only written on the master formula, not subsequent formulas belonging to the same shared group, array, or data table. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
si (Shared Group Index)	Optional attribute to optimize load performance by sharing formulas. When a formula is a shared formula (t value is `shared`) then this value indicates the group to which this particular cell's formula belongs. The first formula in a group of shared formulas is saved in the f element. This is considered the 'master' formula cell. Subsequent cells sharing this formula need not have the formula written in their f element. Instead, the attribute si value for a particular cell is used to figure what the formula expression should be based on the cell's relative location to the master formula cell. A cell is shared only when si is used and t is `shared`. The formula expression for a cell that is specified to be part of a shared formula (and is not the master) shall be ignored, and the master formula shall override. If a master cell of a shared formula range specifies that a particular cell is part of the shared formula range, and that particular cell does not use the si and t attributes to indicate that it is shared, then the particular cell's formula shall override the shared master formula. If this cell occurs in the middle of a range of shared formula cells, the earlier and later formulas shall continue sharing the master formula, and the cell in question shall not share the formula of the master cell formula. Loading and handling of a cell and formula using an si attribute and whose t value is `shared`, located outside the range specified in the master cell associated with the si

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	group, is implementation defined. Master cell references on the same sheet shall not overlap with each other. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
t (Formula Type)	Type of formula. The possible values for this attribute are defined by the ST_CellFormulaType simple type (§18.18.6).
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellFormula) is located in §A.2. *end note*]

18.3.1.41 firstFooter (First Page Footer)

Specifies the contents of the footer used on the first page. Only used when `headerFooter@differentFirst` is '1'. [*Note*:The first logical page in the sheet cannot be printed if the print area is specified to be a range such that it falls outside the first page's scope. *end note*]

The format of a footer is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.42 firstHeader (First Page Header)

Specifies the contents of the header used on the first page. Only used when `headerFooter@differentFirst` is 1. [*Note*: The first logical page in the sheet cannot be printed if the print area is specified to be a range such that it falls outside the first page's scope. *end note*]

The format of a header is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.43 formula (Formula)

The content of this element is a formula whose calculated value specifies the criteria for the conditional formatting rule.

The possible values for this element are defined by the ST_Formula simple type (§18.18.35).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Formula) is located in §A.2. *end note*]

18.3.1.44 formula1 (Formula 1)

The first formula in the Data Validation dropdown. It is used as a bounds for 'between' and 'notBetween' relational operators, and the only formula used for other relational operators (equal, notEqual, lessThan, lessThanOrEqual, greaterThan, greaterThanOrEqual), or for custom or list type data validation. The content can be a formula or a constant or a list series (comma separated values).

The possible values for this element are defined by the ST_Formula simple type (§18.18.35).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Formula) is located in §A.2. *end note*]

18.3.1.45 formula2 (Formula 2)

The second formula in the DataValidation dropdown. It is used as a bounds for 'between' and 'notBetween' relational operators only.

The possible values for this element are defined by the ST_Formula simple type (§18.18.35).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Formula) is located in §A.2. *end note*]

18.3.1.46 headerFooter (Header Footer Settings)

Header and footer settings.

When printed or viewed in page layout view (§18.18.69), each page of a worksheet can have a *page header*, a *page footer*, or both. The headers and footers on odd-numbered pages can differ from those on even-numbered pages, and the headers and footers on the first page can differ from those on odd- and even-numbered pages. In the latter case, the first page is not considered an odd page.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Each header and footer is divided into three areas: a *left section*, a *center section*, and a *right section*, which are specified, respectively, by one or more *left-section-specifier*s, one or more *center-section specifier*s, and one or more *right-section specifier*s. All sections are optional. Each section specifier shall begin with a *formatting code* that indicates whether it is a left-, center-, or right-section specifier. The section specifiers for a header or footer can appear in any order. If a header or footer has multiple section specifiers for the same section, an implementation can concatenate them in the lexical order of their occurrence, into a single, equivalent section specifier. After any such concatenation, the resulting section specifier can contain one or more pieces of text, one or more *formatting codes*, one drawing reference, or a combination thereof. Once specified within a section specifier, a formatting code remains in effect until the end of that section specifier unless toggled or overridden by a subsequent formatting code in that same section specifier.

The values resulting from formatting codes can be localized. An implementation can decide which locales are supported. Even when a locale is not supported, the header and footer text shall be used with only the unsupported formatting being discarded.

Headers and footers are specified using the following elements:

- firstFooter (First Page Footer) (§18.3.1.41)
- firstHeader (First Page Header) (§18.3.1.42)
- oddFooter (Odd Page Footer) (§18.3.1.57)
- oddHeader (Odd Header) (§18.3.1.58)
- evenFooter (Even Page Footer) (§18.3.1.38)
- evenHeader (Even Page Header) (§18.3.1.39)
- drawingHF (Drawing Reference in Header Footer) (§18.3.1.37)

[*Example*:

```
<headerFooter differentFirst="1" differentOddEven="1"> >
  <oddHeader>&amp;R&amp;P</oddHeader>
  <oddFooter>&amp;C&amp;F</oddFooter>
  <evenHeader>&amp;L&amp;P</oddHeader>
  <evenFooter>&amp;L&amp;D&amp;R&amp;T</oddFooter>
  <firstHeader>&amp;CCenter &amp;"-,Bold"Bold
&amp;"-,Regular"Header*U+000A*&amp;D</oddHeader>
</headerFooter>
```

This example shows

- The first page has its own header and footer
- Odd- and even-numbered pages have different headers and footers
- Current page number in the right section of odd-page headers
- Current workbook's file name in the center section of odd-page footers
- Current page number in the left section of even-page headers
- Current date in the left section and the current time in the right section of even-page footers

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- The text “Center **Bold** Header” on the first line of the center section of the first page, and the date on the second line of the center section of that same page
- No footer on the first page

end example]

[*Example*:

An implementation is permitted to concatenate the multiple-like section specifiers in the following:

```
<oddHeader>&amp;LA&amp;CD&amp;RG&amp;LB&amp;CE&amp;RH</oddHeader>
```

treating it as though it was defined instead as follows:

```
<oddHeader>&amp;LAB&amp;CDE&amp;RGH</oddHeader>
```

end example]

The formatting codes are, as follows (for ease of reading, the “amp;“ suffix has been omitted from each & in the entries in the **Formatting Code** column):

Formatting Code	Meaning
&&	The character “&”. [*Example*: “Smith && Sons“ results in “Smith & Sons“ *end example*]
&*font-size*	Size of the text font, where *font-size* is a decimal font size in points. [*Example*: “&18A&36B“ results in “A^B“ *end example*]
&"*font name*,*font type*"	A text font-name string, *font name*, and a text font-type string, *font type*. A *font-name* of “-“ (U+002D) means “no font name is specified”. Both *font-name* and *font-type* can be localized values. Although ISO/IEC 14496-22 permits commas in font family/subfamily/full names, name, and font type, the lexically-first comma in the string is the one recognized as the separating comma. [*Example*: “&"Arial,Regular"…”, “&"Lucida Sans Typewriter,Regular"…”, and “mp;"-,Bold Italic"…“ *end example*]
&"-,Regular"	Regular text format. Toggles bold and italic modes to off.
&A	Current worksheet’s tab name
&B or &"-,Bold"	Bold text format. The next occurrence in a section specifier toggles bold mode, either from off to on, or vice versa. The default mode is off. [*Example*: “ab&Bcd&Bef&Bgh“ results in “ab**cd**ef**gh**“ *end example*]
&C	Center section
&D	Current date
&E	Double-underline text format. The next occurrence in a section specifier toggles double-underline mode, from off to on, or vice versa. The default mode is off. [*Example*: “ab&Ecd&Eef&Egh“ results in “abcdefgh“ *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Formatting Code	Meaning
&F	Current workbook's file name
&G	Drawing object as background. [*Example*: "&L--&G--&R--&G--" results in a drawing with text before and after, in both the left and right sections. *end example*]
&H	Shadow text format. The next occurrence in a section specifier toggles shadow mode, either from off to on, or vice versa. The default mode is off.
&I or &"-,Italic"	Italic text format. The next occurrence in a section specifier toggles italic mode, either from off to on, or vice versa. The default mode is off. [*Example*: "ab&Icd&Ief&Igh" results in "ab*cdefgh*" *end example*]
&K	Text font color • An RGB Color is specified as *RRGGBB* • A Theme Color is specified as *TTSNNN* where *TT* is the theme color Id, *S* is either "+" or "-" of the tint/shade value, and *NNN* is the tint/shade value. [*Example*: "A&KFF0000B&K0070C0C&K01+000D&K07+037E" might result in something like "ABCDE" *end example*]
&L	Left section
&N	Total number of pages. [*Example*: "Page &P of &N" might result in something like "Page 1 of 3" *end example*]
&O	Outline text format. The next occurrence in a section specifier toggles outline mode, either from off to on, or vice versa. The default mode is off.
&P[[+|-]*n*]	Without the optional suffix, the current page number in decimal. [*Example*: See formatting code &N. *end example*] With the optional suffix, the current page number in decimal +/- the decimal number *n*. [*Example*: On Page 1, &P is 1, &P+100 is 101, and &P-5 is -4. *end example*]
&R	Right section
&S	Strikethrough text format. The next occurrence in a section specifier toggles strikethrough mode, either from off to on, or vice versa. The default mode is off. [*Example*: "aa&Sbb&Scc", results in "aa~~bb~~cc". *end example*]
&T	Current time
&U	Single-underline text format. If double-underline mode is on, the next occurrence in a section specifier toggles double-underline mode to off; otherwise, it toggles single-underline mode, from off to on, or vice versa. The default mode is off. [*Example*: "&Ucd&Uef&Ugh" results in "cdefgh" while &Uaa&Ebb&Ucc" results in "aabbcc". *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Formatting Code	Meaning
&X	Superscript text format. The next occurrence in a section specifier toggles superscript mode, either from off to on, or vice versa. The default mode is off. However, superscript and subscript mode cannot be on at the same time. If superscript mode is on, a subsequent subscript format code sets superscript mode off and subscript mode on. [*Example*: "`aa&Xbb&Xcc&Ydd&Yee&Xff&Ygg`" results in "aa^{bb}cc$_{dd}$ee$^{ff}_{gg}$". *end example*]
&Y	Subscript text format. The next occurrence in a section specifier toggles subscript mode, either from off to on, or vice versa. The default mode is off. However, superscript and subscript mode cannot be on at the same time. If subscript mode is on, a subsequent superscript format code sets subscript mode off and superscript mode on. [*Example*: See formatting code &X. *end example*]
&Z	Current workbook's file path

[*Example*:

This example demonstrates "Header" at the top and "Footer" at the bottom of a page.

```
<headerFooter>
  <oddHeader>&amp;CHeader</oddHeader>
  <oddFooter>&amp;CFooter</oddFooter>
</headerFooter>
```

end example]

The tokens in the header & footer elements can be localized. An application can decide which locales are supported. Even when a locale is not supported, the header and footer text shall be loaded, and only the formatting is discarded.

Attributes	Description
alignWithMargins (Align Margins)	Align header footer margins with page margins. When `true`, as left/right margins grow and shrink, the header and footer edges stay aligned with the margins. When `false`, headers and footers are aligned on the paper edges, regardless of margins. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentFirst (Different First Page)	Different first-page header and footer indicator. When `true` then firstHeader and firstFooter specify the first page header and footer values, respectively. If `false` and firstHeader/firstFooter are present, they are ignored. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentOddEven (Different Odd Even	Different odd and even page headers and footers indicator. When `true` then oddHeader/oddFooter and evenHeader/evenFooter specify the page header and

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Header Footer)	footer values for odd and even pages, respectively. If `false` then oddHeader/oddFooter is used, even when evenHeader/evenFooter are present. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
scaleWithDoc (Scale Header & Footer With Document)	Scale header and footer with document scaling. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HeaderFooter) is located in §A.2. *end note*]

18.3.1.47 hyperlink (Hyperlink)

A single hyperlink

Attributes	Description
display (Display String)	Display string, if different from string in string table. This is a property on the hyperlink object, but does not need to appear in the spreadsheet application UI. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id in this sheet's relationships part, expressing the target location of the resource. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
location (Location)	Location within target. If target is a workbook (or this workbook) this shall refer to a sheet and cell or a defined name. Can also be an HTML anchor if target is HTML file. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ref (Reference)	Cell location of hyperlink on worksheet. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
tooltip (Tool Tip)	This is additional text to help the user understand more about the hyperlink. [*Example*: This can be displayed as hover text when the mouse is over the link. *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlink) is located in §A.2. *end note*]

18.3.1.48 hyperlinks (Hyperlinks)

Collection of hyperlinks.

[*Example*:

This example shows a hyperlink in cell A11, with hover text displaying "Search Page". The relationship Id references a relationship from the sheet to the external target resource.

```
<hyperlinks>
  <hyperlink ref="A11" r:id="rId1" tooltip="Search Page"/>
</hyperlinks>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlinks) is located in §A.2. *end note*]

18.3.1.49 iconSet (Icon Set)

Describes an icon set conditional formatting rule.

[*Example*: This example demonstrates the "3Arrows" style of icons. The first icon in the set must be shown if the cell's value is less than the 33rd percentile. The second icon in the set must be shown if the cell's value is less than the 67th percentile, and greater than or equal to the 33rd percentile. The third icon in the set must be shown if the cell's value is greater than or equal to the 67th percentile.

```
<iconSet iconSet="3Arrows">
  <cfvo type="percentile" val="0"/>
  <cfvo type="percentile" val="33"/>
  <cfvo type="percentile" val="67"/>
</iconSet>
```

end example]

Attributes	Description
iconSet (Icon Set)	The icon set to display. The possible values for this attribute are defined by the ST_IconSetType simple type (§18.18.42).
percent (Percent)	Indicates whether the thresholds indicate percentile values, instead of number values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
reverse (Reverse Icons)	If 1, reverses the default order of the icons in this icon set. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showValue (Show Value)	Indicates whether to show the values of the cells on which this icon set is applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_IconSet) is located in §A.2. *end note*]

18.3.1.50 ignoredError (Ignored Error)

A single ignored error for a range of cells.

A cell is considered to have an error condition when it meets one of the conditions specified in the attribute descriptions below. [*Example*: If a cell is formatted as text but contains a numeric value, this is considered to be a potential error because the number won't be treated as a number, for example, in calculations. *end example*]

This is simply a guess by the implementing application, and a recommendation to the user. Cells with the errors specified below might be deliberately defined as such. [*Example*: A cell formatted as text which contains numeric Postal Codes or Order numbers. It is useful to format these cells as text so that leading zeros remain as part of the value instead of being removed. *end example*]

An `<ignoreError>` element is not written in the file unless the user has specifically reviewed the error and decided to keep the cell state as it is, and no longer wishes to be alerted about it for this cell. This can be helpful for the application to decide which errors should be surfaced to the user vs kept quiet because the user doesn't want these to be surfaced (e.g., because they are legitimate cell states).

[*Example*: This example shows that cells A1 and B2 both contain numbers stored as text, and this error has been reviewed and specifically flagged to be no longer surfaced as an error to the user.

```
<ignoredErrors>
  <ignoredError sqref="A1 B2" numberStoredAsText="1"/>
</ignoredErrors>
```

end example]

More than one kind of error can exist on a cell. These flags are not mutually exclusive.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
calculatedColumn (Calculated Column)	Ignore errors when cells contain a value different from a calculated column formula. In other words, for a calculated column, a cell in that column is considered to have an error if its formula is different from the calculated column formula, or doesn't contain a formula at all. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
emptyCellReference (Empty Cell Reference)	Ignore errors when formulas refer to empty cells. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
evalError (Evaluation Error)	Ignore errors when cells contain formulas that result in an error. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
formula (Formula)	Ignore errors when a formula in a region of your worksheet differs from other formulas in the same region. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
formulaRange (Formula Range)	Ignore errors when formulas omit certain cells in a region. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
listDataValidation (List Data Validation)	Ignore errors when a cell's value in a Table does not comply with the Data Validation rules specified. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
numberStoredAsText (Number Stored As Text)	Ignore errors when numbers are formatted as text or are preceded by an apostrophe. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sqref (Sequence of References)	Reference to a range of cells that have this ignored error. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).
twoDigitTextYear (Two Digit Text Year)	Ignore errors when formulas contain text formatted cells with years represented as 2 digits. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
unlockedFormula	Ignore errors when unlocked cells contain formulas.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Unlocked Formula)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_IgnoredError) is located in §A.2. *end note*]

18.3.1.51 ignoredErrors (Ignored Errors)

A collection of ignored errors, by cell range.

[*Note*: The W3C XML Schema definition of this element's content model (CT_IgnoredErrors) is located in §A.2. *end note*]

18.3.1.52 inputCells (Input Cells)

This collection describes each input cell for the scenario.

Attributes	Description
deleted (Deleted)	Input cell was deleted. This input cell shall be present in the file format, but shall not be presented to the user as part of the scenario inputs, nor run as part of the scenario. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
numFmtId (Number Format Id)	This number format Id is used only when displaying the scenario manager input UI, and is used to properly format for display the cached input values (see val attribute) for the scenario. The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
r (Reference)	Cell reference indicating the input cell address. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
undone (Undone)	Cell's deletion was undone. When true the r (reference) value shall not adjust in response to the cell moving due to row / column insert or delete,or cell move. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
val (Value)	Value that should be used for the cell when this scenario is run. val does not need a corresponding data type, the value is put into the cell when the scenario is run.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_InputCells) is located in §A.2. *end note*]

18.3.1.53 is (Rich Text Inline)

This element allows for strings to be expressed directly in the cell definition instead of implementing the shared string table.

[*Example*:

```
<c r="A1">
  <is>
    <t>String</t>
  </is>
</c>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rst) is located in §A.2. *end note*]

18.3.1.54 mergeCell (Merged Cell)

A single merged cell

[*Note*: The W3C XML Schema definition of this element's content model (CT_MergeCell) is located in §A.2. *end note*]

18.3.1.55 mergeCells (Merge Cells)

This collection expresses all the merged cells in the sheet.

[*Example*:

This example shows that three ranges are merged. The formatting and content for the merged range is always stored in the top left cell.

```
<mergeCells>
  <mergeCell ref="C2:F2"/>
  <mergeCell ref="B19:C20"/>
  <mergeCell ref="E19:G19"/>
</mergeCells>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
count (Count)	A count of merged cell collections. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MergeCells) is located in §A.2. *end note*]

18.3.1.56 objectPr (Embedded Object Properties)

This element specifies the visual, positional and cell linkage properties of an embedded object.

[*Example*: The following example demonstrates an embedded object that does not print and that is resized when the cells underlying it are resized:

```
<oleObjects>
  <oleObject … >
    <objectPr print="false" autoLine="false" r:id="rId5">
      <anchor sizeWithCells="true">
        <from> … </from>
        <to> … </to>
      </anchor>
    </objectPr>
  </oleObject>
</oleObjects>
```

end example]

Attributes	Description
altText (Alternative Text)	Specifies alternative text for the object, for use by assistive technologies or applications. [*Example*: `<objectPr altText="Alternate text" … />` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
autoFill (Automatic Fill Flag)	Specifies whether the object's fill formatting is provided automatically by the application. [*Example*: `<objectPr autoFill="false" … />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoLine (Automatic Line Flag)	Specifies whether the object's line formatting is provided automatically by the application. [*Example*: `<objectPr autoLine="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoPict (Automatic Size Flag)	Specifies whether the object's size is formatted automatically by the application. [*Example*: `<objectPr autoPict="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dde (Dynamic Data Exchange Flag)	Specifies whether the object is a Dynamic Data Exchange link. [*Example*: `<objectPr dde="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultSize (Default Size Flag)	Specifies whether the object is at its default size. [*Example*: `<objectPr defaultSize="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
disabled (Disabled Flag)	Specifies whether the object is allowed to run an attached macro. [*Example*: `<objectPr disabled="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Relationship ID to Embedded Object Data) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship which targets the Embedded Object Part containing the embedded object data. The specified relationship shall be of type http://purl.oclc.org/ooxml/officeDocument/relationships/oleObject or the document shall be considered non-conformant. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` targets the part containing the corresponding embedded object information. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
locked (Locked Flag)	Specifies that the object is locked when the sheet is protected. [*Example*: `<objectPr locked="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Custom Function)	Specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] [*Example*: `<objectPr macro="Button1_Click()" … />` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Formula simple type (§18.18.35).
print (Print Flag)	Specifies whether the object is printed when the document is printed. [*Example*: `<objectPr print="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uiObject (UI Object Flag)	Specifies whether the object is a UI-only object. Applications should prevent UI-only objects from being selected and edited in their user interface. [*Example*: `<objectPr uiObject="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectPr) is located in §A.2. *end note*]

18.3.1.57 oddFooter (Odd Page Footer)

Specifies the contents of the page footer used on odd-numbered pages. [*Note*: Odd page(s) in the sheet cannot be printed if the print area is specified to be a range such that it falls outside an odd page's scope. *end note*]

The format of a footer is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.58 oddHeader (Odd Header)

Specifies the contents of the page header used on odd-numbered pages. [*Note*: Odd page(s) in the sheet cannot be printed if the print area is specified to be a range such that it falls outside an odd page's scope. *end note*]

The format of a header is defined in §18.3.1.46.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.59 oleObject (Embedded Object)

Information for an individual embedded object.

Attributes	Description
autoLoad (Auto Load)	Specifies whether the host application for the embedded object shall be called to load the object data automatically when the parent workbook is opened. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dvAspect (Data or View Aspect)	Specifies the desired Data or View Aspect of the object when drawing or getting data The possible values for this attribute are defined by the ST_DvAspect simple type (§18.18.24).
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id of the relationship pointing to the object persistence part. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
link (Embedded Object's Link Moniker)	The embedded object's link moniker. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oleUpdate (Linked Embedded Object Update)	Indicates whether the linked object updates the cached data automatically or only when the container requests an update, only present if the embedded object is linked. The possible values for this attribute are defined by the ST_OleUpdate simple type (§18.18.49).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
progId (Embedded Object ProgId)	ProgId of the embedded object. The possible values for this attribute are defined by the W3C XML Schema string datatype.
shapeId (Shape Id)	Id of the shape this object is associated with. Corresponds with the shape @id in the drawingML part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleObject) is located in §A.2. *end note*]

18.3.1.60 oleObjects (Embedded Objects)

Embedded objects collection in this worksheet.

[*Example*:

This example shows two embedded objects.

```
<oleObjects>
  <oleObject progId="Word.Document.12" shapeId="1025" r:id="rId4"/>
  <oleObject progId="PowerPoint.Show.12" shapeId="1026" r:id="rId5"/>
</oleObjects>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleObjects) is located in §A.2. *end note*]

18.3.1.61 outlinePr (Outline Properties)

Outline properties of the worksheet.

[*Example*: This example indicates that when an outline is applied to data, formatting must be applied to the outline result.

```
<sheetPr>
  <outlinePr applyStyles="1"/>
</sheetPr>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
applyStyles (Apply Styles in Outline)	Flag indicating whether to apply styles in an outline, when outline is applied. Outline styles are described in Styles (§18.8). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showOutlineSymbols (Show Outline Symbols)	Flag indicating whether the sheet has outline symbols visible. This flag shall always be overridden by the showOutlineSymbols attribute on sheetView when there is a conflict. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
summaryBelow (Summary Below)	Flag indicating whether summary rows appear below detail in an outline, when applying an outline. When `true` a summary row is inserted below the detailed data being summarized and a new outline level is established on that row. When `false` a summary row is inserted above the detailed data being summarized and a new outline level is established on that row. Note that toggling this flag on existing outlines requires an update to cell table, specifically, putting the summary functions in the proper rows, and flagging these rows as new outline levels, and possibly resetting their collapsed state. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
summaryRight (Summary Right)	Flag indicating whether summary columns appear to the right of detail in an outline, when applying an outline. When `true`, a summary column is inserted to the right of the detailed data being summarized and a new outline level is established on that column. When `false` a summary column is inserted to the left of the detailed data being summarized and a new outline level is established on that column. Note that toggling this flag on existing outlines requires an update to cell table, specifically, putting the summary functions in the proper columns, and flagging these columns as new outline levels, and possibly resetting their collapsed state. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OutlinePr) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.3.1.62 pageMargins (Page Margins)

Page margins for a sheet or a custom sheet view.

[*Example*:

```
<pageMargins left="0.7" right="0.7" top="0.75" bottom="0.75" header="0.3"
  footer="0.3"/>
```

end example]

Attributes	Description
bottom (Bottom Page Margin)	Bottom Page Margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
footer (Footer Page Margin)	Footer Page Margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
header (Header Page Margin)	Header Page Margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
left (Left Page Margin)	Left Page Margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
right (Right Page Margin)	Right page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
top (Top Page Margin)	Top Page Margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageMargins) is located in §A.2. *end note*]

18.3.1.63 pageSetup (Page Setup Settings)

Page setup settings for the worksheet.

[*Example*: The following example shows the pageSetup element for ISO A0 paper, printed in black and white, with graphics:

```
<pageSetup blackAndWhite="true" draft="false" paperHeight="1189mm"
paperWidth="841mm" />
```

end example]

Attributes	Description
blackAndWhite (Black And White)	Print black and white. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
cellComments (Print Cell Comments)	This attribute specifies how to print cell comments. The possible values for this attribute are defined by the ST_CellComments simple type (§18.18.5).
copies (Number Of Copies)	Number of copies to print. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
draft (Draft)	Print without graphics. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
errors (Print Error Handling)	Specifies how to print cell values for cells with errors. The possible values for this attribute are defined by the ST_PrintError simple type (§18.18.60).
firstPageNumber (First Page Number)	Page number for first printed page. If no value is specified, then 'automatic' is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
fitToHeight (Fit To Height)	Number of vertical pages to fit on. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
fitToWidth (Fit To Width)	Number of horizontal pages to fit on. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
horizontalDpi (Horizontal DPI)	Horizontal print resolution of the device. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Id)	Relationship Id of the devMode printer settings part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
orientation (Orientation)	Orientation of the page. The possible values for this attribute are defined by the ST_Orientation simple type (§18.18.50).
pageOrder (Page Order)	Order of printed pages. The possible values for this attribute are defined by the ST_PageOrder simple type (§18.18.51).
paperHeight (Paper Height)	Height of custom paper as a number followed by a unit identifier. [*Example*: `297mm`, `11in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
paperSize (Paper Size)	Paper size 1 = Letter paper (8.5 in. by 11 in.) 2 = Letter small paper (8.5 in. by 11 in.) 3 = Tabloid paper (11 in. by 17 in.) 4 = Ledger paper (17 in. by 11 in.) 5 = Legal paper (8.5 in. by 14 in.) 6 = Statement paper (5.5 in. by 8.5 in.) 7 = Executive paper (7.25 in. by 10.5 in.) 8 = A3 paper (297 mm by 420 mm) 9 = A4 paper (210 mm by 297 mm) 10 = A4 small paper (210 mm by 297 mm) 11 = A5 paper (148 mm by 210 mm) 12 = B4 paper (250 mm by 353 mm) 13 = B5 paper (176 mm by 250 mm) 14 = Folio paper (8.5 in. by 13 in.) 15 = Quarto paper (215 mm by 275 mm) 16 = Standard paper (10 in. by 14 in.) 17 = Standard paper (11 in. by 17 in.) 18 = Note paper (8.5 in. by 11 in.) 19 = #9 envelope (3.875 in. by 8.875 in.) 20 = #10 envelope (4.125 in. by 9.5 in.) 21 = #11 envelope (4.5 in. by 10.375 in.) 22 = #12 envelope (4.75 in. by 11 in.) 23 = #14 envelope (5 in. by 11.5 in.) 24 = C paper (17 in. by 22 in.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	25 = D paper (22 in. by 34 in.) 26 = E paper (34 in. by 44 in.) 27 = DL envelope (110 mm by 220 mm) 28 = C5 envelope (162 mm by 229 mm) 29 = C3 envelope (324 mm by 458 mm) 30 = C4 envelope (229 mm by 324 mm) 31 = C6 envelope (114 mm by 162 mm) 32 = C65 envelope (114 mm by 229 mm) 33 = B4 envelope (250 mm by 353 mm) 34 = B5 envelope (176 mm by 250 mm) 35 = B6 envelope (176 mm by 125 mm) 36 = Italy envelope (110 mm by 230 mm) 37 = Monarch envelope (3.875 in. by 7.5 in.). 38 = 6 3/4 envelope (3.625 in. by 6.5 in.) 39 = US standard fanfold (14.875 in. by 11 in.) 40 = German standard fanfold (8.5 in. by 12 in.) 41 = German legal fanfold (8.5 in. by 13 in.) 42 = ISO B4 (250 mm by 353 mm) 43 = Japanese double postcard (200 mm by 148 mm) 44 = Standard paper (9 in. by 11 in.) 45 = Standard paper (10 in. by 11 in.) 46 = Standard paper (15 in. by 11 in.) 47 = Invite envelope (220 mm by 220 mm) 50 = Letter extra paper (9.275 in. by 12 in.) 51 = Legal extra paper (9.275 in. by 15 in.) 52 = Tabloid extra paper (11.69 in. by 18 in.) 53 = A4 extra paper (236 mm by 322 mm) 54 = Letter transverse paper (8.275 in. by 11 in.) 55 = A4 transverse paper (210 mm by 297 mm) 56 = Letter extra transverse paper (9.275 in. by 12 in.) 57 = SuperA/SuperA/A4 paper (227 mm by 356 mm) 58 = SuperB/SuperB/A3 paper (305 mm by 487 mm) 59 = Letter plus paper (8.5 in. by 12.69 in.) 60 = A4 plus paper (210 mm by 330 mm) 61 = A5 transverse paper (148 mm by 210 mm) 62 = JIS B5 transverse paper (182 mm by 257 mm) 63 = A3 extra paper (322 mm by 445 mm) 64 = A5 extra paper (174 mm by 235 mm) 65 = ISO B5 extra paper (201 mm by 276 mm) 66 = A2 paper (420 mm by 594 mm) 67 = A3 transverse paper (297 mm by 420 mm) 68 = A3 extra transverse paper (322 mm by 445 mm) 69 = Japanese Double Postcard (200 mm x 148 mm) 70 = A6 (105 mm x 148 mm) 71 = Japanese Envelope Kaku #2 72 = Japanese Envelope Kaku #3

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	73 = Japanese Envelope Chou #3 74 = Japanese Envelope Chou #4 75 = Letter Rotated (11in x 8 1/2 11 in) 76 = A3 Rotated (420 mm x 297 mm) 77 = A4 Rotated (297 mm x 210 mm) 78 = A5 Rotated (210 mm x 148 mm) 79 = B4 (JIS) Rotated (364 mm x 257 mm) 80 = B5 (JIS) Rotated (257 mm x 182 mm) 81 = Japanese Postcard Rotated (148 mm x 100 mm) 82 = Double Japanese Postcard Rotated (148 mm x 200 mm) 83 = A6 Rotated (148 mm x 105 mm) 84 = Japanese Envelope Kaku #2 Rotated 85 = Japanese Envelope Kaku #3 Rotated 86 = Japanese Envelope Chou #3 Rotated 87 = Japanese Envelope Chou #4 Rotated 88 = B6 (JIS) (128 mm x 182 mm) 89 = B6 (JIS) Rotated (182 mm x 128 mm) 90 = (12 in x 11 in) 91 = Japanese Envelope You #4 92 = Japanese Envelope You #4 Rotated 93 = PRC 16K (146 mm x 215 mm) 94 = PRC 32K (97 mm x 151 mm) 95 = PRC 32K(Big) (97 mm x 151 mm) 96 = PRC Envelope #1 (102 mm x 165 mm) 97 = PRC Envelope #2 (102 mm x 176 mm) 98 = PRC Envelope #3 (125 mm x 176 mm) 99 = PRC Envelope #4 (110 mm x 208 mm) 100 = PRC Envelope #5 (110 mm x 220 mm) 101 = PRC Envelope #6 (120 mm x 230 mm) 102 = PRC Envelope #7 (160 mm x 230 mm) 103 = PRC Envelope #8 (120 mm x 309 mm) 104 = PRC Envelope #9 (229 mm x 324 mm) 105 = PRC Envelope #10 (324 mm x 458 mm) 106 = PRC 16K Rotated 107 = PRC 32K Rotated 108 = PRC 32K(Big) Rotated 109 = PRC Envelope #1 Rotated (165 mm x 102 mm) 110 = PRC Envelope #2 Rotated (176 mm x 102 mm) 111 = PRC Envelope #3 Rotated (176 mm x 125 mm) 112 = PRC Envelope #4 Rotated (208 mm x 110 mm) 113 = PRC Envelope #5 Rotated (220 mm x 110 mm) 114 = PRC Envelope #6 Rotated (230 mm x 120 mm) 115 = PRC Envelope #7 Rotated (230 mm x 160 mm) 116 = PRC Envelope #8 Rotated (309 mm x 120 mm) 117 = PRC Envelope #9 Rotated (324 mm x 229 mm) 118 = PRC Envelope #10 Rotated (458 mm x 324 mm)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Note*: To maximize interoperability, implementers should restrict the content of this attribute to enumerations present in the above list. Additional values may be used, but interoperability will only be possible via mutual agreement between implementers. *end note*] When values not present in the above list are used, the behavior is implementation-defined. When paperHeight and paperWidth are specified, paperSize should be ignored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
paperWidth (Paper Width)	Width of custom paper as a number followed by a unit identifier. [*Example*: `21cm`, `8.5in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
scale (Print Scale)	Print scaling. This attribute is restricted to values ranging from 10 to 400. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] This setting is overridden when fitToWidth and/or fitToHeight are in use. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
useFirstPageNumber (Use First Page Number)	Use firstPageNumber value for first page number, and do not auto number the pages. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
usePrinterDefaults (Use Printer Defaults)	Use the printer's defaults settings for page setup values and don't use the default values specified in the schema. [*Example*: If dpi is not present or specified in the XML, the application must not assume 600dpi as specified in the schema as a default and instead must let the printer specify the default dpi. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
verticalDpi (Vertical DPI)	Vertical print resolution of the device. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageSetup) is located in §A.2. *end note*]

18.3.1.64 pageSetup (Chart Sheet Page Setup)

This element provides page setup properties for chart sheets.

[*Example*: The following example shows the pageSetup element for ISO A0 paper, printed in black and white, with graphics:

```
<pageSetup blackAndWhite="true" draft="false" paperHeight="1189mm"
paperWidth="841mm" />
```

end example]

Attributes	Description
blackAndWhite (Black And White)	Print black and white. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
copies (Number Of Copies)	Number of copies to print. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
draft (Draft)	Print draft quality. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
firstPageNumber (First Page Number)	Page number for first printed page. If no value is specified, then 'automatic' is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
horizontalDpi (Horizontal DPI)	Horizontal print resolution of the device. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
id (Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id of the devMode printer settings part. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
orientation (Orientation)	Orientation of the page. The possible values for this attribute are defined by the ST_Orientation simple type (§18.18.50).
paperHeight (Paper Height)	Height of custom paper as a number followed by a unit identifier. [*Example*: `297mm`, `11in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
paperSize (Paper Size)	1 = Letter paper (8.5 in. by 11 in.) 2 = Letter small paper (8.5 in. by 11 in.) 3 = Tabloid paper (11 in. by 17 in.) 4 = Ledger paper (17 in. by 11 in.) 5 = Legal paper (8.5 in. by 14 in.) 6 = Statement paper (5.5 in. by 8.5 in.) 7 = Executive paper (7.25 in. by 10.5 in.) 8 = A3 paper (297 mm by 420 mm) 9 = A4 paper (210 mm by 297 mm) 10 = A4 small paper (210 mm by 297 mm) 11 = A5 paper (148 mm by 210 mm) 12 = B4 paper (250 mm by 353 mm) 13 = B5 paper (176 mm by 250 mm) 14 = Folio paper (8.5 in. by 13 in.) 15 = Quarto paper (215 mm by 275 mm) 16 = Standard paper (10 in. by 14 in.) 17 = Standard paper (11 in. by 17 in.) 18 = Note paper (8.5 in. by 11 in.) 19 = #9 envelope (3.875 in. by 8.875 in.) 20 = #10 envelope (4.125 in. by 9.5 in.) 21 = #11 envelope (4.5 in. by 10.375 in.) 22 = #12 envelope (4.75 in. by 11 in.) 23 = #14 envelope (5 in. by 11.5 in.) 24 = C paper (17 in. by 22 in.) 25 = D paper (22 in. by 34 in.) 26 = E paper (34 in. by 44 in.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	27 = DL envelope (110 mm by 220 mm) 28 = C5 envelope (162 mm by 229 mm) 29 = C3 envelope (324 mm by 458 mm) 30 = C4 envelope (229 mm by 324 mm) 31 = C6 envelope (114 mm by 162 mm) 32 = C65 envelope (114 mm by 229 mm) 33 = B4 envelope (250 mm by 353 mm) 34 = B5 envelope (176 mm by 250 mm) 35 = B6 envelope (176 mm by 125 mm) 36 = Italy envelope (110 mm by 230 mm) 37 = Monarch envelope (3.875 in. by 7.5 in.). 38 = 6 3/4 envelope (3.625 in. by 6.5 in.) 39 = US standard fanfold (14.875 in. by 11 in.) 40 = German standard fanfold (8.5 in. by 12 in.) 41 = German legal fanfold (8.5 in. by 13 in.) 42 = ISO B4 (250 mm by 353 mm) 43 = Japanese double postcard (200 mm by 148 mm) 44 = Standard paper (9 in. by 11 in.) 45 = Standard paper (10 in. by 11 in.) 46 = Standard paper (15 in. by 11 in.) 47 = Invite envelope (220 mm by 220 mm) 50 = Letter extra paper (9.275 in. by 12 in.) 51 = Legal extra paper (9.275 in. by 15 in.) 52 = Tabloid extra paper (11.69 in. by 18 in.) 53 = A4 extra paper (236 mm by 322 mm) 54 = Letter transverse paper (8.275 in. by 11 in.) 55 = A4 transverse paper (210 mm by 297 mm) 56 = Letter extra transverse paper (9.275 in. by 12 in.) 57 = SuperA/SuperA/A4 paper (227 mm by 356 mm) 58 = SuperB/SuperB/A3 paper (305 mm by 487 mm) 59 = Letter plus paper (8.5 in. by 12.69 in.) 60 = A4 plus paper (210 mm by 330 mm) 61 = A5 transverse paper (148 mm by 210 mm) 62 = JIS B5 transverse paper (182 mm by 257 mm) 63 = A3 extra paper (322 mm by 445 mm) 64 = A5 extra paper (174 mm by 235 mm) 65 = ISO B5 extra paper (201 mm by 276 mm) 66 = A2 paper (420 mm by 594 mm) 67 = A3 transverse paper (297 mm by 420 mm) 68 = A3 extra transverse paper (322 mm by 445 mm) [*Note*: To maximize interoperability, implementers should restrict the content of this attribute to enumerations present in the above list. Additional values may be used, but interoperability will only be possible via mutual agreement between implementers. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	When values not present in the above list are used, the behavior is implementation-defined. When paperHeight and paperWidth are specified, paperSize should be ignored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
paperWidth (Paper Width)	Width of custom paper as a number followed by a unit identifier. [*Example*: `21cm`, `8.5in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
useFirstPageNumber (Use First Page Number)	Use firstPageNumber value for first page number, and do not auto number the pages. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
usePrinterDefaults (Use Printer Defaults)	Use the printer's defaults settings for page setup values and don't use the default values specified in the schema. [*Example*: If dpi is not present or specified in the XML, the application must not assume 600dpi as specified in the schema as a default and instead must let the printer specify the default dpi. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
verticalDpi (Vertical DPI)	Vertical print resolution of the device. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CsPageSetup) is located in §A.2. *end note*]

18.3.1.65 pageSetUpPr (Page Setup Properties)

Page setup properties of the worksheet

Attributes	Description
autoPageBreaks (Show Auto Page Breaks)	Flag indicating whether the sheet displays Automatic Page Breaks. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fitToPage (Fit To	Flag indicating whether the Fit to Page print option is enabled.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Page)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageSetUpPr) is located in §A.2. *end note*]

18.3.1.66 pane (View Pane)

Worksheet view pane

Attributes	Description
activePane (Active Pane)	The pane that is active. The possible values for this attribute are defined by the ST_Pane simple type (§18.18.52).
state (Split State)	Indicates whether the pane has horizontal / vertical splits, and whether those splits are frozen. The possible values for this attribute are defined by the ST_PaneState simple type (§18.18.53).
topLeftCell (Top Left Visible Cell)	Location of the top left visible cell in the bottom right pane (when in Left-To-Right mode). The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
xSplit (Horizontal Split Position)	Horizontal position of the split, in 1/20th of a point; 0 (zero) if none. If the pane is frozen, this value indicates the number of columns visible in the top pane. The possible values for this attribute are defined by the W3C XML Schema double datatype.
ySplit (Vertical Split Position)	Vertical position of the split, in 1/20th of a point; 0 (zero) if none. If the pane is frozen, this value indicates the number of rows visible in the left pane. The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pane) is located in §A.2. *end note*]

18.3.1.67 picture (Background Image)

Background sheet image.

[*Example*:

```
<picture r:id="rId1"/>
```

end example]

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id pointing to the image part. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetBackgroundPicture) is located in §A.2. *end note*]

18.3.1.68 pivotArea (Pivot Area)

Rule describing a PivotTable selection.

Attributes	Description
axis (Axis)	The region of the PivotTable to which this rule applies. The possible values for this attribute are defined by the ST_Axis simple type (§18.18.1).
cacheIndex (Cache Index)	Flag indicating whether any indexes refer to fields or items in the Pivot cache and not the view. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
collapsedLevelsAreSubtotals (Collapsed Levels Are Subtotals)	Flag indicating if collapsed levels/dimensions are considered subtotals. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataOnly (Data Only)	Flag indicating whether only the data values (in the data area of the view) for an item selection are selected and does not include the item labels. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
field (Field Index)	Index of the field that this selection rule refers to. The possible values for this attribute are defined by the W3C XML Schema int datatype.
fieldPosition (Field Position)	Position of the field within the axis to which this rule applies. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
grandCol (Include Column Grand Total)	Flag indicating whether the column grand total is included. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
grandRow (Include Row Grand Total)	Flag indicating whether the row grand total is included. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
labelOnly (Labels Only)	Flag indicating whether only the item labels for an item selection are selected and does not include the data values (in the data area of the view). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
offset (Offset Reference)	A Reference that specifies a subset of the selection area. Points are relative to the top left of the selection area. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
outline (Outline)	Flag indicating whether the rule refers to an area that is in outline mode. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
type (Rule Type)	Indicates the type of selection rule. The possible values for this attribute are defined by the ST_PivotAreaType simple type (§18.18.58).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotArea) is located in §A.2. *end note*]

18.3.1.69 pivotSelection (PivotTable Selection)

A collection of PivotTable structure selections. A PivotTable structure selection is a way of specifying what cells in the PivotTable are selected. Instead of specifying cell addresses in a sqref, a particular area or structure within the PivotTable is specified. In this way there is semantic meaning regarding what is selected, rather than simply a list of cell or ranges contained in the selection. Typically fields on the row or column axis are selected.

[*Example*: For example, the innermost field (Product SubCategory) is selected in this PivotTable:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	A	B	C
1			
2		State	(All)
3		City	(All)
4			
5			Column Labels
6			2001
7			3
8			July
9		Row Labels	Sum of Sales Amount
10		Bikes	209652.9046
11		Mountain Bikes	64424.81
12		Mountain-100 Black, 38	3374.99
13		Mountain-100 Black, 42	3374.99
14		Mountain-100 Black, 44	13499.96
15		Mountain-100 Black, 48	3374.99
16		Mountain-100 Silver, 38	6799.98
17		Mountain-100 Silver, 42	6799.98
18		Mountain-100 Silver, 44	16999.95
19		Mountain-100 Silver, 48	10199.97
20		Road Bikes	145228.0946
21		Road-150 Red, 44	25047.89
22		Road-150 Red, 48	42939.24
23		Road-150 Red, 52	21469.62
24		Road-150 Red, 56	25047.89
25		Road-150 Red, 62	28626.16
26		Road-650 Black, 44	699.0982
27		Road-650 Black, 52	
28		Road-650 Black, 62	699.0982
29		Road-650 Red, 44	699.0982
30		Road-650 Red, 48	
31		Road-650 Red, 52	
32		Road-650 Red, 58	
33		Road-650 Red, 60	
34		Grand Total	209652.9046

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The corresponding pivotSelection XML should look like this:

```
<pivotSelection pane="bottomRight" showHeader="1" axis="axisRow" dimension="2"
  activeRow="11" activeCol="1" previousRow="11" previousCol="1" click="1"
  r:id="rId1">
```


©ISO/IEC 2016 – All rights reserved

```
    <pivotArea dataOnly="0" labelOnly="1" fieldPosition="0">
      <references count="1">
        <reference field="9" count="0"/>
      </references>
    </pivotArea>
  </pivotSelection>
```

axis indicates that this selection is on the row axis, dimension indicates the field level within the row axis that is selected (zero-based index), activeCol and activeRow respectively indicate where in the grid the selection is located, and reference field indicates to which particular field the selection corresponds.

end example]

Attributes	Description
activeCol (Active Column)	The column (zero-based) of active cell for structure selection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
activeRow (Active Row)	The row (zero-based) of active cell for structure selection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
axis (Axis)	Axis of the PivotTable on which this selection lies. The possible values for this attribute are defined by the ST_Axis simple type (§18.18.1).
click (Click Count)	Number of clicks for this structure selection. For some selection combinations, subsequent clicks on the same target area cycles the actual selection through some variances. Therefore number of clicks on the selection shall be recorded, if it is desirable to restore this state of the selection cycle on load. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
count (Selection Count)	Number of selections for the structure selection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
data (Data Selection)	Flag indicating whether the structure selection is for data only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dimension (Dimension)	Indicates the field level within the axis that is selected (zero-based index). The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
extendable (Extendable)	Flag indicating whether the structure selection can have additional selections added to it. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship Id pointing to the particular PivotTable Part corresponding to this selection. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
label (Label)	Flag indicating whether the structure selection is for labels only (e.g., a grand total row is selected). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
max (Maximum)	The maximum line the structure selection contains. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
min (Minimum)	The minimum line the structure selection contains. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
pane (Pane)	The pane to which this PivotTable structure selection belongs. The possible values for this attribute are defined by the ST_Pane simple type (§18.18.52).
previousCol (Previous Column Selection)	1-based index to the column immediately left of the structure selection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
previousRow (Previous Row)	1-based index to the row immediately above the structure selection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
showHeader (Show Header)	Flag indicating whether selection toggle from data only to header only to both is enabled. `False` means disabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
start (Start)	The line the structure selection begins (zero-based). This is the line clicked to initiate the structure selection.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotSelection) is located in §A.2. *end note*]

18.3.1.70 printOptions (Print Options)

Print options for the sheet. Printer-specific settings are stored separately in the Printer Settings part as defined in §15.2.15.

Attributes	Description
gridLines (Print Grid Lines)	Used in conjunction with gridLinesSet. If both gridLines and gridlinesSet are `true`, then grid lines shall print. Otherwise, they shall not (i.e., one or both have `false` values). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
gridLinesSet (Grid Lines Set)	Used in conjunction with gridLines. If both gridLines and gridLinesSet are `true`, then grid lines shall print. Otherwise, they shall not (i.e., one or both have `false` values). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
headings (Print Headings)	Print row and column headings. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
horizontalCentered (Horizontal Centered)	Center on page horizontally when printing. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
verticalCentered (Vertical Centered)	Center on page vertically when printing. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PrintOptions) is located in §A.2. *end note*]

18.3.1.71 protectedRange (Protected Range)

A specified range to be protected. Ranges listed here are protected only when the sheet protection is ON and the cell is flagged as being locked. If no password is specified here, then read/write permissions are automatically given to all users, regardless of additional security descriptor information. In other words, the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

security descriptor information (specific types of access) at the user level is only applied if a password for this range is specified.

When a password is to be hashed and stored in this element, it shall be hashed as defined below, starting from a UTF-16LE encoded string value. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

When a password is specified, then users not listed specifically as having access should be prompted with a password. If that user supplies the correct password, then they can edit the range or cell in question. This protection is optional and can be ignored by applications that choose not to support this functionality.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>algorithmName (Cryptographic Algorithm Name)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value.

The following values are reserved:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>MD2</td><td>Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>
[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:
<pre>< … algorithmName="SHA-1"
  hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The algorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. end example]
The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
</td></tr>
<tr><td>hashValue
(Password Hash Value)</td><td>
Specifies the hash value for the password required to edit this range. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.
[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:
<pre><… AlgorithmName="SHA-1"
  hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The hashValue attribute value of 9oN7nWkCAyEZib1RomSJTjmPpCY= specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of SHA-1) and that the resulting has value must be 9oN7nWkCAyEZib1RomSJTjmPpCY= for the protection to be disabled. end example]
The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
name (Name)	Range title. This is used as a descriptor, not as a named range definition. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The saltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The spinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	hashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sqref (Sequence of References)	The range to be protected. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ProtectedRange) is located in §A.2. *end note*]

18.3.1.72 protectedRanges (Protected Ranges)

This collection specifies all protected ranges on this worksheet.

[*Example*:

This example demonstrates that A1:C5 have been protected, with no password specified.

```
<protectedRanges>
  <protectedRange sqref="A1:C5" name="Range1"/>
</protectedRanges>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ProtectedRanges) is located in §A.2. *end note*]

18.3.1.73 row (Row)

The element expresses information about an entire row of a worksheet, and contains all cell definitions for a particular row in the worksheet.

[*Example*:

This row expresses information about row 2 in the worksheet, and contains 3 cell definitions.

```
<row r="2" spans="2:12">
  <c r="C2" s="1">
    <f>PMT(B3/12,B4,-B5)</f>
    <v>672.68336574300008</v>
  </c>
  <c r="D2">
    <v>180</v>
  </c>
```

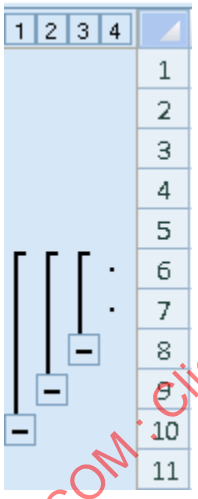
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <c r="E2">
      <v>360</v>
    </c>
  </row>
```

end example]

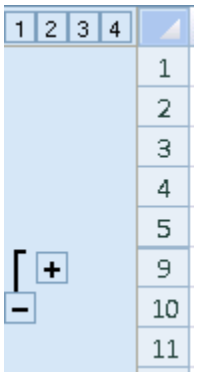
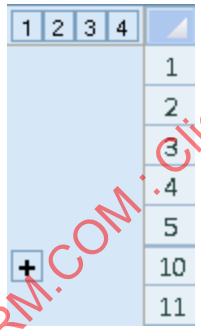
Attributes	Description

collapsed
(Collapsed)
	1 if the rows 1 level of outlining deeper than the current row are in the collapsed outline state. It means that the rows which are 1 outline level deeper (numerically higher value) than the current row are currently hidden due to a collapsed outline state.
	
	It is possible for collapsed to be `false` and yet still have the rows in question hidden. This can be achieved by having a lower outline level collapsed, thus hiding all the child rows.
	
	[*Example*:
	This example shows 3 levels of outlining:
	
	 
	
	In the XML must be:
	`<sheetData>`
	`  <row r="6" outlineLevel="3"/>`
	`  <row r="7" outlineLevel="3"/>`
	`  <row r="8" outlineLevel="2"/>`
	`  <row r="9" outlineLevel="1"/>`
	`</sheetData>`
	
	end example]
	
	[*Example*:
	This example shows the same outline feature, with the middle level collapsed:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	
	In the XML must be:
	`<sheetData>`
	`<row r="6" hidden="1" outlineLevel="3"/>`
	`<row r="7" hidden="1" outlineLevel="3"/>`
	`<row r="8" hidden="1" outlineLevel="2"/>`
	`<row r="9" outlineLevel="1" collapsed="1"/>`
	`</sheetData>`
	
	end example]
	
	[*Example*:
	This example shows the same outline feature as above, where both the middle and lowest level are collapsed:
	
	 
	In the XML must be:
	
	`<sheetData>`
	`<row r="6" hidden="1" outlineLevel="3"/>`
	`<row r="7" hidden="1" outlineLevel="3"/>`
	`<row r="8" hidden="1" outlineLevel="2"/>`
	`<row r="9" hidden="1" outlineLevel="1" collapsed="1"/>`
	`<row r="10" collapsed="1"/>`
	`</sheetData>`
	
	Note that in this case, if the lowest level were expanded, the middle level would remain

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	collapsed due to collapsed being `true` on row 9. *end example*] See description of outlinePr element's summaryBelow and summaryRight attributes for detailed information. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
customFormat (Custom Format)	1 if the row style should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
customHeight (Custom Height)	1 if the row height has been manually set. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hidden (Hidden)	1 if the row is hidden, e.g., due to a collapsed outline or by manually selecting and hiding a row. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ht (Row Height)	Row height measured in point size. There is no margin padding on row height. The possible values for this attribute are defined by the W3C XML Schema double datatype.
outlineLevel (Outline Level)	Outlining level of the row, when outlining is on. See description of outlinePr element's summaryBelow and summaryRight attributes for detailed information. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
ph (Show Phonetic)	1 if the row should show phonetic. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Row Index)	Row index. Indicates to which row in the sheet this <row> definition corresponds. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
s (Style Index)	Index to style record for the row (only applied if customFormat attribute is '1') The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

spans (Spans)	Optimization only, and not required. Specifies the range of non-empty columns (in the format X:Y) for the block of rows to which the current row belongs. To achieve the optimization, span attribute values in a single block should be the same.
	
	There are 16 rows per block, beginning with the first row.
	
	[*Note*: this is an optimization, and is purely optional. Different span values within the same row block is allowed. Not writing the span value at all is also allowed. *end note*]
	
	Blank rows are not required to write out span values.
	
	[*Example*: If cells F8, E9, and D10 have data in them and the rest of the sheet is empty, then for those three rows (8,9, and 10), the spans value should each be "4:6":
	
	`<sheetData>`
	`<row r="8" spans="4:6">`
	`<c r="F8">`
	`<v>1</v>`
	`</c>`
	`</row>`
	`<row r="9" spans="4:6">`
	`<c r="E9">`
	`<v>2</v>`
	`</c>`
	`</row>`
	`<row r="10" spans="4:6">`
	`<c r="D10">`
	`<v>3</v>`
	`</c>`
	`</row>`
	`</sheetData>`
	
	If cells A1 and J10 have data in them and the rest of the sheet is empty, then the rows should be written like this:
	
	`<sheetData>`
	`<row r="1" spans="1:10">`
	`<c r="A1">`
	`<v>1</v>`
	`</c>`
	`</row>`
	`<row r="10" spans="1:10">`
	`<c r="J10">`
	`<v>2</v>`
	`</c>`
	`</row>`
	`</sheetData>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_CellSpans simple type (§18.18.9).
thickBot (Thick Bottom)	1 if any cell in the row has a medium or thick bottom border, or if any cell in the row directly below the current row has a thick top border. When `true` and customHeight is `false`, this flag means that the row height has been adjusted higher by .75 points of the normal style font height. This also means that if the row no longer contains these borders, then the height is automatically re-adjusted down. This adjustment is in addition to any adjustment of height due to thickTop. Medium borders are these enumeration values from the Styles Part: • mediumDashDotDot • slantDashDot • mediumDashDot • mediumDashed • medium • • Thick borders are these enumeration values from the Styles Part: • thick • double The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
thickTop (Thick Top Border)	`True` if the row has a medium or thick top border, or if any cell in the row directly above the current row has a thick bottom border. When `true` and customHeight is `false`, this flag means that the row height has been adjusted higher by .75 points of the normal style font height. This also means that if the row no longer contains these borders, then the height is automatically re-adjusted down. This adjustment is in addition to any adjustment of height due to thickBot. Medium borders are these enumeration values from the Styles Part: • mediumDashDotDot • slantDashDot • mediumDashDot • mediumDashed • medium Thick borders are these enumeration values from the Styles Part: • thick • double

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Row) is located in §A.2. *end note*]

18.3.1.74 rowBreaks (Horizontal Page Breaks (Row))

A collection of row breaks (§18.3.1.3).

[*Example*: This example shows a break inserted at cell B25:

```
<rowBreaks count="1" manualBreakCount="1">
  <brk id="24" max="16383" man="1"/>
</rowBreaks>
```

end example]

Attributes	Description
count (Page Break Count)	Number of breaks in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
manualBreakCount (Manual Break Count)	Number of manual breaks in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageBreak) is located in §A.2. *end note*]

18.3.1.75 scenario (Scenario)

An individual scenario description. [*Note*: See parent element for an example. *end note*]

Attributes	Description
comment (Scenario Comment)	Comment for this scenario, rich text not supported. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
count (Changing Cell Count)	Number of input cells. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
hidden (Hidden Scenario)	Scenario is hidden when the sheet is protected and 'edit scenarios' is not enabled in sheet protection options. If the scenario is marked as hidden but sheet protection options specify to allow editing scenarios, then the scenario shall not be hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
locked (Scenario Locked)	Scenario is locked for editing when the sheet is protected. If sheet is protected and "edit scenarios" is enabled, then this setting is ignored. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Scenario Name)	Scenario's name (user input). Shall be unique for the worksheet. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
user (User Name)	Name of user who last changed the scenario. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scenario) is located in §A.2. *end note*]

18.3.1.76 scenarios (Scenarios)

A collection of Scenarios. A scenario is a named what-if model that includes variable cells linked together by one or more formulas.

[*Example*: For example, you might want to compare best-case and worst-case scenarios for sales in a coffee shop, based on the number of cups of coffee sold in a week.

```
<scenarios current="1" show="0" sqref="G4 G6 G7 G8">
  <scenario name="Best Case" locked="1" count="3" user="anonymous"
    comment="Created on 6/9/2006_x000a_Modified on 6/9/2006">
    <inputCells r="D5" val="151" numFmtId="37"/>
    <inputCells r="D9" val="226"/>
    <inputCells r="D13" val="126"/>
  </scenario>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <scenario name="Worst Case" locked="1" count="3" user="anonymous"
      comment="Created on 6/9/2006">
      <inputCells r="D5" val="50" numFmtId="37"/>
      <inputCells r="D9" val="40"/>
      <inputCells r="D13" val="30"/>
    </scenario>
  </scenarios>
```

end example]

Attributes	Description
current (Current Scenario)	Zero-based index to current scenario selected. Can correspond to selection UI. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
show (Last Shown Scenario)	Zero-based index to last shown scenario. Indicates which scenario was last selected by the user to be run/shown. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sqref (Sequence of References)	Range or sequence of cells used for scenario results summary. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scenarios) is located in §A.2. *end note*]

18.3.1.77 securityDescriptor (Security Descriptor)

Optional setting to specify the relative security descriptor. The security descriptor defines user accounts who may edit this range without providing a password to access the range.

[*Note*: The format of a securityDescriptor is application defined; however, it is recommended that the following format be used for interoperability between implementations: `username@domain`. This format follows the form of `addr-spec` as defined in RFC 822, Standard for ARPA Internet Text Messages. *end note*]

[*Example*: This example demonstrates two user accounts in the security descriptor attribute:

```
  <protectedRanges>
    <protectedRange sqref="A1:C5" name="Range1">
      <securityDescriptor>user1@iso.org</securityDescriptor>
      <securityDescriptor>user2@iso.org</securityDescriptor>
    </protectedRange>
  </protectedRanges>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

18.3.1.78 selection (Selection)

Worksheet view selection.

Attributes	Description
activeCell (Active Cell Location)	Location of the active cell. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
activeCellId (Active Cell Index)	0-based index of the range reference (in the array of references listed in sqref) containing the active cell. Only used when the selection in sqref is not contiguous. Therefore, this value needs to be aware of the order in which the range references are written in sqref. When this value is out of range then activeCell can be used. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
pane (Pane)	The pane to which this selection belongs. The possible values for this attribute are defined by the ST_Pane simple type (§18.18.52).
sqref (Sequence of References)	Range of the selection. Can be non-contiguous set of ranges. The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Selection) is located in §A.2. *end note*]

18.3.1.79 sheetCalcPr (Sheet Calculation Properties)

This element contains calculation properties for the worksheet.

Attributes	Description
fullCalcOnLoad (Full Calculation On Load)	Indicates whether the application should do a full calculate on load due to contents on this sheet. After load and successful calc, the application shall set this value to false. Set this to true when the application should calculate the workbook on load. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetCalcPr) is located in §A.2. *end note*]

18.3.1.80 sheetData (Sheet Data)

This collection represents the cell table itself. This collection expresses information about each cell, grouped together by rows in the worksheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetData) is located in §A.2. *end note*]

18.3.1.81 sheetFormatPr (Sheet Format Properties)

Sheet formatting properties.

Attributes	Description
baseColWidth (Base Column Width)	Specifies the number of characters of the maximum digit width of the normal style's font. This value does not include margin padding or extra padding for gridlines. It is only the number of characters. See defaultColWidth description in this section for details on calculating this value. See the col element description, particularly the width attribute description, for more information on what is meant by "maximum digit width". The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
customHeight (Custom Height)	'True' if defaultRowHeight value has been manually set, or is different from the default value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultColWidth (Default Column Width)	Default column width measured as the number of characters of the maximum digit width of the normal style's font. If the user has not set this manually, then it can be calculated: defaultColWidth = baseColumnWidth + {margin padding (2 pixels on each side, totalling 4 pixels)} + {gridline (1pixel)} If the user has set this manually, then there is no calculation, and simply a value is specified. The possible values for this attribute are defined by the W3C XML Schema double datatype.
defaultRowHeight (Default Row Height)	Default row height measured in point size. Optimization so we don't have to write the height on all rows. This can be written out if most rows have custom height, to achieve the optimization.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	When the row height of all rows in a sheet is the default value, then that value is written here, and customHeight is not set. If a few rows have a different height, that information is written directly on each row. However, if most or all of the rows in the sheet have the same height, but that height isn't the default height, then that height value should be written here (as an optimization), and the customHeight flag should also be set. In this case, all rows having this height do not need to express the height, only rows whose height differs from this value need to be explicitly expressed. The possible values for this attribute are defined by the W3C XML Schema double datatype.
outlineLevelCol (Column Outline Level)	Highest number of outline levels for columns in this sheet. These values shall be in synch with the actual sheet outline levels. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
outlineLevelRow (Maximum Outline Row)	Highest number of outline level for rows in this sheet. These values shall be in synch with the actual sheet outline levels. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
thickBottom (Thick Bottom Border)	'True' if rows have a thick bottom border by default. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
thickTop (Thick Top Border)	'True' if rows have a thick top border by default. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
zeroHeight (Hidden By Default)	'True' if rows are hidden by default. This setting is an optimization used when most rows of the sheet are hidden. In this case, instead of writing out every row and specifying hidden, it is much shorter to only write out the rows that are not hidden, and specify here that rows are hidden by default, and only not hidden if specified. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetFormatPr) is located in §A.2. *end note*]

18.3.1.82 sheetPr (Sheet Properties)

Sheet-level properties.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
codeName (Code Name)	Specifies a stable name of the sheet, which should not change over time, and does not change from user input. This name should be used by code to reference a particular sheet. The possible values for this attribute are defined by the W3C XML Schema string datatype.
enableFormatConditionsCalculation (Enable Conditional Formatting Calculations)	Flag indicating whether the conditional formatting calculations shall be evaluated. If set to `false`, then the min/max values of color scales or databars or threshold values in Top N rules shall not be updated. Essentially the conditional formatting "calc" is off. This is useful when conditional formats are being set programmatically at runtime, recalculation of the conditional formatting does not need to be done until the program execution has finished setting all the conditional formatting properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
filterMode (Filter Mode)	Flag indicating whether the worksheet has one or more autofilters or advanced filters on. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
published (Published)	Flag indicating whether the worksheet is published. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
syncHorizontal (Synch Horizontal)	Flag indicating whether this worksheet is horizontally synced to the synchRef anchor point. When true and scroll location is missing from the window properties, the window view shall be scrolled to the horizontal (row) aspect of the synchRef value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
syncRef (Synch Reference)	Anchor point for worksheet's window. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
syncVertical (Synch Vertical)	Flag indicating whether this worksheet is vertically synced to the synchRef anchor point. When true and scroll location is missing from the window properties, the window view shall be scrolled to the vertical (column) aspect of the synchRef value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
transitionEntry (Transition Formula Entry)	Flag indicating whether the Transition Formula Entry (Lotus compatibility) option is enabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
transitionEvaluation (Transition Formula Evaluation)	Flag indicating whether the Transition Formula Evaluation (Lotus compatibility) option is enabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetPr) is located in §A.2. *end note*]

18.3.1.83 sheetPr (Chart Sheet Properties)

This element specifies chart sheet properties.

Attributes	Description
codeName (Code Name)	Specifies a stable name of the sheet, which should not change over time, and does not change from user input. This name should be used by code to reference a particular sheet. The possible values for this attribute are defined by the W3C XML Schema string datatype.
published (Published)	Flag indicating whether the chart sheet is published. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartsheetPr) is located in §A.2. *end note*]

18.3.1.84 sheetProtection (Chart Sheet Protection)

This collection expresses the chart sheet protection options to enforce when the chart sheet is protected.

Attributes	Description
algorithmName (Cryptographic Algorithm Name)	Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value. The following values are reserved:

Value	Algorithm
MD2	Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td></td><td>breaks. *end note*]</td></tr><tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used. [*Note*: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. *end note*]</td></tr><tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr><tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr></table> [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `< … algorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The algorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
content (Contents)	When true prevents users from making changes to items that are part of the chart, such as data series, axes, and legends. The chart continues to reflect changes made to its source data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

hashValue (Password Hash Value)	Specifies the hash value for the password required to edit this chartsheet. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.
	
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements:
	
	`<… algorithmName="SHA-1"`
	`hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hashValue attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password shall be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of `SHA-1`) and that the resulting has value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

objects (Objects Locked)	When true prevents users from making changes to graphic objects— including shapes, text boxes, and controls— unless you unlock the specific objects before you protect the chart sheet. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The saltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The spinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartsheetProtection) is located in §A.2. *end note*]

18.3.1.85 sheetProtection (Sheet Protection Options)

This collection expresses the sheet protection options to enforce when the sheet is protected.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This example demonstrates that the sheet is protected, objects and scenarios can be edited, cell formatting is allowed,and selection of locked cells is not allowed:

```
<sheetProtection sheet="1" objects="1" scenarios="1" formatCells="0"
  selectLockedCells="1"/>
```

end example]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>algorithmName (Cryptographic Algorithm Name)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value.

The following values are reserved:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>MD2</td><td>Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table>
[Example: Consider an Office Open XML document with the following information stored in one of its protection elements:
<pre>< … algorithmName="SHA-1"
hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" /></pre>
The algorithmName attribute value of “SHA-1” specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. end example]
The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
</td></tr>
<tr><td>autoFilter (AutoFilter Locked)</td><td>
If 1 or true then AutoFilters should not be allowed to operate when the sheet is protected.
If 0 or false then AutoFilters should be allowed to operate when the sheet is protected.
The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
</td></tr>
<tr><td>deleteColumns (Delete Columns Locked)</td><td>
If 1 or true then deleting columns should not be allowed when the sheet is protected.
If 0 or false then deleting columns should be allowed when the sheet is protected.
The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
</td></tr>
<tr><td>deleteRows (Delete Rows Locked)</td><td>
If 1 or true then deleting rows should not be allowed when the sheet is protected.
If 0 or false then deleting rows should be allowed when the sheet is protected.
The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
</td></tr>
<tr><td>formatCells (Format Cells Locked)</td><td>
If 1 or true then formatting cells should not be allowed when the sheet is protected.
If 0 or false then formatting cells should be allowed when the sheet is protected.
The possible values for this attribute are defined by the W3C XML Schema boolean
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
formatColumns (Format Columns Locked)	If 1 or true then formatting columns should not be allowed when the sheet is protected. If 0 or false then formatting columns should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
formatRows (Format Rows Locked)	If 1 or true then formatting rows should not be allowed when the sheet is protected. If 0 or false then formatting rows should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

hashValue (Password Hash Value)	Specifies the hash value for the password required to edit this worksheet. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.
	
	If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook.
	
	[*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements:
	
	`<… algorithmName="SHA-1"`
	`hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />`
	
	The hashValue attribute value of `9oN7nWkCAyEZib1RomSJTjmPpCY=` specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of `SHA-1`) and that the resulting hash value must be `9oN7nWkCAyEZib1RomSJTjmPpCY=` for the protection to be disabled. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.

insertColumns (Insert Columns Locked)	If 1 or true then inserting columns should not be allowed when the sheet is protected. If 0 or false then inserting columns should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
insertHyperlinks (Insert Hyperlinks	If 1 or true then inserting hyperlinks should not be allowed when the sheet is protected.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Locked)	If 0 or false then inserting hyperlinks should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
insertRows (Insert Rows Locked)	If `1` or `true` then inserting rows should not be allowed when the sheet is protected. If 0 or false then inserting rows should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
objects (Objects Locked)	If `1` or `true` then editing of objects should not be allowed when the sheet is protected. If `0` or `false` then objects are allowed to be edited when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pivotTables (Pivot Tables Locked)	If `1` or `true` then PivotTables should not be allowed to operate when the sheet is protected. If 0 or false then PivotTables should be allowed to operate when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack"). If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The saltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
scenarios (Scenarios Locked)	If `1` or `true` then Scenarios should not be edited when the sheet is protected. If `0` or `false` then Scenarios are allowed to be edited when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
selectLockedCells (Select Locked Cells Locked)	If `1` or `true` then selection of locked cells should not be allowed when the sheet is protected. If 0 or false then selection of locked cells should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
selectUnlockedCells (Select Unlocked Cells Locked)	If `1` or `true` then selection of unlocked cells should not be allowed when the sheet is protected. If 0 or false then selection of unlocked cells should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sheet (Sheet Locked)	If `1` or `true` then the sheet is protected. If `0` or `false` then the sheet is not protected. The value of this attribute dictates whether the other attributes of sheetProtection should be applied. If 1 or true then the other attributes of sheetProtection should be applied. If 0 or false then the other attributes of sheetProtection should not be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sort (Sort Locked)	If `1` or `true` then sorting should not be allowed when the sheet is protected. If `0` or `false` then sorting should be allowed when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The spinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetProtection) is located in §A.2. *end note*]

18.3.1.86 sheetView (Chart Sheet View)

This element specifies a chart sheet view. [*Note*: See sheetView (§18.3.1.87) for an example. *end note*]

Attributes	Description
tabSelected (Sheet Tab Selected)	Flag indicating whether the sheet tab is selected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
workbookViewId (Workbook View Id)	Zero-based index of this workbook view, pointing to a workbookView element in the bookViews collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomScale (Window Zoom Scale)	Window zoom magnification, representing percent values. This attribute is restricted to values ranging from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100%

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… 400 - 400% *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomToFit (Zoom To Fit)	Flag indicating whether chart sheet is zoom to fit window. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartsheetView) is located in §A.2. *end note*]

18.3.1.87 sheetView (Worksheet View)

A single sheet view definition. When more than one sheet view is defined in the file, it means that when opening the workbook, each sheet view corresponds to a separate window within the spreadsheet application, where each window is showing the particular sheet containing the same workbookViewId value, the last sheetView definition is loaded, and the others are discarded. When multiple windows are viewing the same sheet, multiple sheetView elements (with corresponding workbookView entries) are saved.

Attributes	Description
colorId (Color Id)	Index to the color value for row/column text headings and gridlines. This is an 'index color value' (ICV) rather than rgb value. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
defaultGridColor (Default Grid Color)	Flag indicating that the consuming application should use the default grid lines color (system dependent). Overrides any color specified in colorId. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rightToLeft (Right To Left)	Flag indicating whether the sheet is in 'right to left' display mode. When in this mode, Column A is on the far right, Column B ;is one column left of Column A, and so on. Also, information in cells is displayed in the Right to Left format. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showFormulas (Show Formulas)	Flag indicating whether this sheet should display formulas. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
showGridLines (Show Grid Lines)	Flag indicating whether this sheet should display gridlines. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showOutlineSymbols (Show Outline Symbols)	Flag indicating whether the sheet has outline symbols visible. This flag shall always override SheetPr element's outlinePr child element whose attribute is named showOutlineSymbols when there is a conflict. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRowColHeaders (Show Headers)	Flag indicating whether the sheet should display row and column headings. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRuler (Show Ruler)	Show the ruler in page layout view (§18.18.69).. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showWhiteSpace (Show White Space)	Flag indicating whether page layout view (§18.18.69) shall display margins. False means do not display left, right, top (header), and bottom (footer) margins (even when there is data in the header or footer). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showZeros (Show Zero Values)	Flag indicating whether the window should show 0 (zero) in cells containing zero value. When false, cells with zero value appear blank instead of showing the number zero. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tabSelected (Sheet Tab Selected)	Flag indicating whether this sheet is selected. When only 1 sheet is selected and active, this value should be in synch with the activeTab value. In case of a conflict, the Start Part setting wins and sets the active sheet tab. Multiple sheets can be selected, but only one sheet shall be active at one time. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
topLeftCell (Top Left Visible Cell)	Location of the top left visible cell Location of the top left visible cell in the bottom right pane (when in Left-to-Right mode). The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
view (View Type)	Indicates the view type. [*Note*: Although this Standard is for a file format, occasionally, guidance is given regarding intent in dealing with things outside that file format, such as the rendering of documents to a screen or printer. *end note*] The possible values for this attribute are defined by the ST_SheetViewType simple type (§18.18.69).
windowProtection (Window Protection)	Flag indicating whether the panes in the window are locked due to workbook protection. This is an option when the workbook structure is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
workbookViewId (Workbook View Index)	Zero-based index of this workbook view, pointing to a workbookView element in the bookViews collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomScale (Zoom Scale)	Window zoom magnification for current view representing percent values. This attribute is restricted to values ranging from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] The current view can be normal view, page break preview, or page layout view (§18.18.69). The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomScaleNormal (Zoom Scale Normal View)	Zoom magnification to use when in normal view (§18.18.69), representing percent values. This attribute is restricted to values ranging from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100% …

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	400 - 400% *end example*] Applies for worksheets only; zero implies the automatic setting. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomScalePageLayoutView (Zoom Scale Page Layout View)	Zoom magnification to use when in page layout view (§18.18.69), representing percent values. This attribute is restricted to values ranging from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] Applies for worksheets only; zero implies the automatic setting. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
zoomScaleSheetLayoutView (Zoom Scale Page Break Preview)	Zoom magnification to use when in page break preview (§18.18.69), representing percent values. This attribute is restricted to values ranging from 10 to 400. Horizontal & Vertical scale together. [*Example*: 10 - 10% 20 - 20% … 100 - 100% … 400 - 400% *end example*] Applies for worksheet only; zero implies the automatic setting. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetView) is located in §A.2. *end note*]

18.3.1.88 sheetViews (Sheet Views)

Worksheet views collection.

[*Example*:

This example shows one sheet view definition. The definition indicates that the current sheet is the active/selected sheet, and that there is a split pane applied to the view. This definition also indicates for each of the four window panes of the split which cell is the active cell for that pane.

```
<sheetViews>
  <sheetView tabSelected="1" workbookViewId="0">
    <pane xSplit="2310" ySplit="2070" topLeftCell="C1"
      activePane="bottomRight"/>
    <selection/>
    <selection pane="bottomLeft" activeCell="A6" sqref="A6"/>
    <selection pane="topRight" activeCell="C1" sqref="C1"/>
    <selection pane="bottomRight" activeCell="E13" sqref="E13"/>
  </sheetView>
</sheetViews>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetViews) is located in §A.2. *end note*]

18.3.1.89 sheetViews (Chart Sheet Views)

This element specifies chart sheet views.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartsheetViews) is located in §A.2. *end note*]

18.3.1.90 smartTags (Smart Tags)

This collection expresses all smart tags associated with cells on this sheet. There can be multiple smart tags associated with a particular cell, and many cells with smart tags for a given worksheet.

[*Example*:

This example shows three smart tags, each one associated with a unique cell on the worksheet.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<smartTags>
  <cellSmartTags r="A1">
    <cellSmartTag type="0"/>
  </cellSmartTags>
  <cellSmartTags r="B1">
    <cellSmartTag type="0"/>
  </cellSmartTags>
  <cellSmartTags r="B2">
    <cellSmartTag type="0"/>
  </cellSmartTags>
</smartTags>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTags) is located in §A.2. *end note*]

18.3.1.91 sortCondition (Sort Condition)

Sort condition. When more than one sortCondition is specified, the first condition is applied first, then the second condition is applied, and so on.

Attributes	Description
customList (Custom List)	Sort by a custom list. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
descending (Descending)	Sort descending. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dxfId (Format Id)	Format Id when sortBy=cellColor or fontColor The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
iconId (Icon Id)	Zero-based index of an icon in an icon set. The absence of this attribute means "no icon" The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
iconSet (Icon Set)	Icon set index when sortBy=icon. The possible values for this attribute are defined by the ST_IconSetType simple type (§18.18.42).
ref (Reference)	Column/Row that this sort condition applies to. This shall be contained within the ref in CT_SortState.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sortBy (Sort By)	Type of sort. The possible values for this attribute are defined by the ST_SortBy simple type (§18.18.72).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SortCondition) is located in §A.2. *end note*]

18.3.1.92 sortState (Sort State)

This collection preserves the AutoFilter sort state.

[*Example*: This example shows a sort which is case-sensitive, descending sort. While the range of data to sort is B4:E8, the range to sort by is B4:B8.

```
<sortState caseSensitive="1" ref="B4:E8">
  <sortCondition descending="1" ref="B4:B8"/>
</sortState>
```

end example]

Attributes	Description
caseSensitive (Case Sensitive)	Flag indicating whether or not the sort is case-sensitive. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
columnSort (Sort by Columns)	Flag indicating whether or not to sort by columns. Only applies to ranges that don't have AutoFilter applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ref (Sort Range)	The whole range of data to sort (not just the sort-by column). The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sortMethod (Sort Method)	Strokes or PinYin sort method. Applies only to these application UI languages: • Chinese Simplified • Chinese Traditional • Japanese • For these languages, alternate sort methods can be selected, affecting how the data is sorted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_SortMethod simple type (§18.18.73).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SortState) is located in §A.2. *end note*]

18.3.1.93 tabColor (Sheet Tab Color)

Background color of the sheet tab.

Attributes	Description
auto (Automatic)	A boolean value indicating the color is automatic and system color dependent. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indexed (Index)	Indexed color value. Only used for backwards compatibility. References a color in indexedColors. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rgb (Alpha Red Green Blue Color Value)	Standard Alpha Red Green Blue color value (ARGB). The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
theme (Theme Color)	A zero-based index into the `<clrScheme>` collection (§20.1.6.2), referencing a particular `<sysClr>` or `<srgbClr>` value expressed in the Theme part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
tint (Tint)	Specifies the tint value applied to the color. If tint is supplied, then it is applied to the RGB value of the color to determine the final color applied. The tint value is stored as a double from -1.0 .. 1.0, where -1.0 means 100% darken and 1.0 means 100% lighten. Also, 0.0 means no change. In loading the RGB value, it is converted to HLS where HLS values are (0..HLSMAX), where HLSMAX is currently 255. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here are some examples of how to apply tint to color: **If (tint < 0)** Lum' = Lum * (1.0 + tint) For example: Lum = 200; tint = -0.5; Darken 50% Lum' = 200 * (0.5) => 100 For example: Lum = 200; tint = -1.0; Darken 100% (make black) Lum' = 200 * (1.0-1.0) => 0 **If (tint > 0)** Lum' = Lum * (1.0-tint) + (HLSMAX – HLSMAX * (1.0-tint)) For example: Lum = 100; tint = 0.75; Lighten 75% Lum' = 100 * (1-.75) + (HLSMAX – HLSMAX*(1-.75)) = 100 * .25 + (255 – 255 * .25) = 25 + (255 – 63) = 25 + 192 = 217 For example: Lum = 100; tint = 1.0; Lighten 100% (make white) Lum' = 100 * (1-1) + (HLSMAX – HLSMAX*(1-1)) = 100 * 0 + (255 – 255 * 0) = 0 + (255 – 0) = 255 *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.2. *end note*]

18.3.1.94 tablePart (Table Part)

A single Table Part reference.

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This relationship Id is used to locate a particular table definition part. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePart) is located in §A.2. *end note*]

18.3.1.95 tableParts (Table Parts)

This collection expresses a relationship Id pointing to every table on this sheet.

[*Example*: This example indicates that the current sheet has two tables, and their definitions can be found by locating the appropriate relationships from the sheet:

```
<tableParts count="2">
  <tablePart r:id="rId1"/>
  <tablePart r:id="rId2"/>
</tableParts>
```

end example]

Attributes	Description
count (Count)	A count of table elements in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableParts) is located in §A.2. *end note*]

18.3.1.96 v (Cell Value)

This element expresses the value contained in a cell. If the cell contains a string, then this value is an index into the shared string table, pointing to the actual string value. Otherwise, the value of the cell is expressed directly in this element. Cells containing formulas express the last calculated result of the formula in this element.

For applications not wanting to implement the shared string table, an 'inline string' can be expressed in an <is> element under <c> (instead of a <v> element under <c>),in the same way a string would be expressed in the shared string table. [*Note*: See <is> for an example. *end note*]

[*Example*: In this example, cell B4 contains the number "360", cell C4 contains the local date and time 22 November 1976, 08:30, and cell C5 contains the 1900 date system serial date-time for the date-time in cell C4.

```
<c r="B4">
  <v>360</v>
</c>
<c r="C4" t="d">
  <v>1976-11-22T08:30</v>
</c>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<c r="C5">
  <f>C4</f>
   <v>28086.3541666667</v>
</c>
```

end example]

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.3.1.97 webPublishItem (Web Publishing Item)

This element represents information for a single item or object which can be published to HTML.

Attributes	Description
autoRepublish (Automatically Publish)	Automatically publish this item every time the workbook is saved. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
destinationFile (Destination File Name)	Destination file name. Indicates where to save the HTML publish file. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
divId (Destination Bookmark)	Destination bookmark. Identifies a specific section in the published HTML file when a subset of the workbook is published to HTML. Each item that has been published from a workbook is written to a unique element in HTML. On re-publishing a particular item from the workbook, only that item's corresponding content is updated. Therefore each publish item corresponds to a unique element. It is possible to add new publish items to an existing published page, and it is possible to re-publish individual items without republishing the entire workbook. The possible values for this attribute are defined by the ST_Xstring simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
id (Id)	This is a unique number "nnnnn" of the webPublishItem. This value is used to generate the divId and styleId values. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sourceObject (Source Object Name)	Source object name (required for sourceType = pivotTable, query, or label). The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sourceRef (Source Id)	Source range (required for sourceType = 'range'). The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sourceType (Web Source Type)	Type of web source (or objects to publish). The possible values for this attribute are defined by the ST_WebSourceType simple type (§18.18.92).
title (Title)	HTML title of published item. [*Example*: This value can appear in the web browser window's title bar. *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPublishItem) is located in §A.2. *end note*]

18.3.1.98 webPublishItems (Web Publishing Items)

This represents a listing of individual objects in this workbook that have been published (to HTML).

When one of these objects is selected to be published, just the object is published to HTML, not the entire workbook contents.

[*Example*: This example shows two items which have been previously selected for publishing. One is a range (A6:C6), the other is a chart, named "Chart 1".

```
<webPublishItems count="2">
  <webPublishItem id="11289" divId="Views_11289" sourceType="range"
    sourceRef="A6:C6" destinationFile="D:\Publish.htm" published="0"/>
  <webPublishItem id="6433" divId="Views_6433" sourceType="chart"
    sourceObject="Chart 1" destinationFile="D:\Publish.mht" published="0"/>
</webPublishItems>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

Attributes	Description
count (Web Publishing Items Count)	Number of items. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPublishItems) is located in §A.2. *end note*]

18.3.1.99 worksheet (Worksheet)

This is the root element of Worksheet parts within a SpreadsheetML document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Worksheet) is located in §A.2. *end note*]

18.3.2 AutoFilter Settings

The following subclause defines the settings which can be specified as part of an AutoFilter definition. An *AutoFilter* temporarily hides rows based on a filter criteria, which is applied column by column to a table of data in the worksheet.

18.3.2.1 colorFilter (Color Filter Criteria)

This element specifies the color to filter by and whether to use the cell's fill or font color in the filter criteria. If the cell's font or fill color does not match the color specified in the criteria, the rows corresponding to those cells are hidden from view.

[*Example*:

```
<filterColumn colId="1">
  <colorFilter dxfId="0" cellColor="0"/>
</filterColumn>
```

end example]

Attributes	Description
cellColor (Filter By Cell Color)	Flag indicating whether or not to filter by the cell's fill color. 1 indicates to filter by cell fill. 0 indicates to filter by the cell's font color. For rich text in cells, if the color specified appears in the cell at all, it shall be included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
dxfId (Differential Format Record Id)	Id of differential format record (dxf) in the Styles Part which expresses the color value to filter by. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorFilter) is located in §A.2. *end note*]

18.3.2.2 customFilter (Custom Filter Criteria)

A custom AutoFilter specifies an operator and a value. There can be at most two customFilters specified, and in that case the parent element specifies whether the two conditions are joined by 'and' or 'or'. For any cells whose values do not meet the specified criteria, the corresponding rows shall be hidden from view when the filter is applied.

[*Example*:

```
<customFilters and="1">
  <customFilter operator="greaterThanOrEqual" val="0.2"/>
  <customFilter operator="lessThanOrEqual" val="0.5"/>
</customFilters>
```

end example]

Attributes	Description
operator (Filter Comparison Operator)	Operator used by the filter comparison. The possible values for this attribute are defined by the ST_FilterOperator simple type (§18.18.31).
val (Top or Bottom Value)	Top or bottom value used in the filter criteria. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomFilter) is located in §A.2. *end note*]

18.3.2.3 customFilters (Custom Filters)

When there is more than one custom filter criteria to apply (an 'and' or 'or' joining two criteria), then this element groups the customFilter elements together.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
and (And)	Flag indicating whether the two criteria have an "and" relationship. 1 indicates "and", 0 indicates "or". The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomFilters) is located in §A.2. *end note*]

18.3.2.4 dateGroupItem (Date Grouping)

This collection is used to express a group of dates or times which are used in an AutoFilter criteria. [*Note*: See parent element for an example. *end note*] Values are always written in the calendar type of the first date encountered in the filter range, so that all subsequent dates, even when formatted or represented by other calendar types, can be correctly compared for the purposes of filtering.

Attributes	Description
dateTimeGrouping (Date Time Grouping)	Grouping level. The possible values for this attribute are defined by the ST_DateTimeGrouping simple type (§18.18.22).
day (Day)	Day (1-31) The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
hour (Hour)	Hour (0-23) The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
minute (Minute)	Minute (0-59) The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
month (Month)	Month (1-12) The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
second (Second)	Second (0-59) The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
year (Year)	Year (4 digits)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DateGroupItem) is located in §A.2. *end note*]

18.3.2.5 dynamicFilter (Dynamic Filter)

This collection specifies dynamic filter criteria. These criteria are considered dynamic because they can change, either with the data itself (e.g., "above average") or with the current system date (e.g., show values for "today"). For any cells whose values do not meet the specified criteria, the corresponding rows shall be hidden from view when the filter is applied.

[*Example*:

```
<filterColumn colId="0">
  <dynamicFilter type="today"/>
</filterColumn>
```

end example]

Attributes	Description
maxValIso (Max ISO Value)	A maximum value for dynamic filter. maxValIso shall be required for today, yesterday, tomorrow, nextWeek, thisWeek, lastWeek, nextMonth, thisMonth, lastMonth, nextQuarter, thisQuarter, lastQuarter, nextYear, thisYear, lastYear, and yearToDate. The above criteria are based on a value range; that is, if today's date is September 22nd, then the range for thisWeek is the values greater than or equal to September 17 and less than September 24. In the thisWeek range, the lower value is expressed valIso. The higher value is expressed using maxValIso. These dynamic filters shall not require valIso or maxValIso: Q1, Q2, Q3, Q4, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11 and M12. The above criteria shall not specify the range using valIso and maxValIso because Q1 always starts from M1 to M3, and M1 is always January. These types of dynamic filters shall use valIso and shall not use maxValIso: • aboveAverage and belowAverage The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
type (Dynamic filter type)	Dynamic filter type, e.g., “today” or “nextWeek”. The possible values for this attribute are defined by the ST_DynamicFilterType simple type (§18.18.26).
val (Value)	A minimum numeric value for dynamic filter. (See description of valIso to understand when val is required.) The possible values for this attribute are defined by the W3C XML Schema double datatype.
valIso (ISO Value)	A minimum value for dynamic filter. (See description of maxValIso to understand when val/valIso is required.) Only these types of dynamic filters use numeric data, and therefore shall use val and shall not use valIso: • aboveAverage and belowAverage The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.

[*Note*: The W3C XML Schema definition of this element’s content model (CT_DynamicFilter) is located in §A.2. *end note*]

18.3.2.6 filter (Filter)

This element expresses a filter criteria value.

[*Example*:

```
<filters>
  <filter val="0.316588716"/>
  <filter val="0.667439395"/>
  <filter val="0.823086999"/>
</filters>
```

end example]

Attributes	Description
val (Filter Value)	Filter value used in the criteria. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Filter) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.3.2.7 filterColumn (AutoFilter Column)

The filterColumn collection identifies a particular column in the AutoFilter range and specifies filter information that has been applied to this column. If a column in the AutoFilter range has no criteria specified, then there is no corresponding filterColumn collection expressed for that column.

Attributes	Description
colId (Filter Column Data)	Zero-based index indicating the AutoFilter column to which this filter information applies. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
hiddenButton (Hidden AutoFilter Button)	Flag indicating whether the AutoFilter button for this column is hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showButton (Show Filter Button)	Flag indicating whether an application intended for editing Office Open XML documents should show filtering interface elements on this cell. [*Example*: When a cell containing a filter button is merged with another cell, the filter button can be hidden, and not drawn. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FilterColumn) is located in §A.2. *end note*]

18.3.2.8 filters (Filter Criteria)

When multiple values are chosen to filter by, or when a group of date values are chosen to filter by, this element groups those criteria together.

[*Example*:

```
<filters>
  <dateGroupItem year="2006" month="1" day="2" dateTimeGrouping="day"/>
  <dateGroupItem year="2005" month="1" day="2" dateTimeGrouping="day"/>
</filters>
```

end example]

Attributes	Description
blank (Filter by Blank)	Flag indicating whether to filter by blank. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
calendarType (Calendar Type)	Calendar type for date grouped items. Used to interpret the values in dateGroupItem. This is the calendar type used to evaluate all dates in the filter column, even when those dates are not using the same calendar system / date formatting. The possible values for this attribute are defined by the ST_CalendarType simple type (§22.9.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Filters) is located in §A.2. *end note*]

18.3.2.9 iconFilter (Icon Filter)

This element specifies the icon set and particular icon within that set to filter by. For any cells whose icon does not match the specified criteria, the corresponding rows shall be hidden from view when the filter is applied.

[*Example*:

```
<filterColumn colId="3">
  <iconFilter iconSet="3Arrows" iconId="0"/>
</filterColumn>
```

end example]

Attributes	Description
iconId (Icon Id)	Zero-based index of an icon in an icon set. The absence of this attribute means "no icon" The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
iconSet (Icon Set)	Specifies which icon set is used in the filter criteria. The possible values for this attribute are defined by the ST_IconSetType simple type (§18.18.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_IconFilter) is located in §A.2. *end note*]

18.3.2.10 top10 (Top 10)

This element specifies the top N (percent or number of items) to filter by.

[*Example*: This example filters the first column by the top 10 percent of the values in that column. For all cells in the column whose value falls outside the top 10 percent of the value in that column, the rows corresponding to those cells are hidden from the view. In this example, there are 6 cells in the range, containing 1, 2, 3, 4, 5, 6 respectively.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<filterColumn colId="0">
  <top10 percent="1" val="5" filterVal="6"/>
</filterColumn
```

end example]

Attributes	Description
filterVal (Filter Value)	The actual cell value in the range which is used to perform the comparison for this filter. The possible values for this attribute are defined by the W3C XML Schema double datatype.
percent (Filter by Percent)	Flag indicating whether or not to filter by percent value of the column. A false value filters by number of items. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
top (Top)	Flag indicating whether or not to filter by top order. A false value filters by bottom order. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
val (Top or Bottom Value)	Top or bottom value to use as the filter criteria. [*Example*: "Filter by Top 10 Percent" or "Filter by Top 5 Items". *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Top10) is located in §A.2. *end note*]

18.4 Shared String Table

String values may be stored directly inside spreadsheet cell elements (§18.3.1.4); however, storing the same value inside multiple cell elements can result in very large worksheet Parts, possibly resulting in performance degradation. The Shared String Table is an indexed list of string values, shared across the workbook, which allows implementations to store values only once.

[*Example*: Consider for example a workbook summarizing information for cities within various countries. There can be a column for the name of the country, a column for the name of each city in that country, and a column containing the data for each city. In this case the country name is repetitive, being duplicated in many cells. *end example*] In many cases the repetition is extensive, and significant savings are realized by making use of a shared string table when saving the workbook. When displaying text in the spreadsheet, the cell table will just contain an index into the string table as the value of a cell, instead of the full string.

The shared string table is permitted to contain all the necessary information for displaying the string: the text, formatting properties, and phonetic properties (for East Asian languages).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Most strings in a workbook have formatting applied at the cell level, that is, the entire string in the cell has the same formatting applied. In these cases, the formatting for the cell is stored in the styles part, and the string for the cell can be stored in the shared strings table. In this case, the strings stored in the shared strings table are very simple text elements.

[*Example*:

```
<sst xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main"
  count="8" uniqueCount="4">
  <si>
    <t>United States</t>
  </si>
  <si>
    <t>Seattle</t>
  </si>
  <si>
    <t>Denver</t>
  </si>
  <si>
    <t>New York</t>
  </si>
</sst>
```

In the above example we can see that the string table is just a collection of string items that consist of simple text elements. Note that any numeric data in the workbook is not shown in the shared string table. *end example*]

Some strings in the workbook can have formatting applied at a level that is more granular than the cell level. For instance, specific characters within the string can be bolded, have coloring, italicizing, etc. In these cases, the formatting is stored along with the text in the string table, and is treated as a unique entry in the table.

[*Example*:

```
<sst xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main"
  count="8" uniqueCount="4">
  <si>
    <r>
      <t xml:space="preserve">United </t>
    </r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <r>
        <rPr>
          <sz val="11"/>
          <color rgb="FFFF0000"/>
          <rFont val="Calibri"/>
          <family val="2"/>
          <scheme val="minor"/>
        </rPr>
        <t>States</t>
      </r>
    </si>
    <si>
      <t>Seattle</t>
    </si>
    <si>
      <t>Denver</t>
    </si>
    <si>
      <t>New York</t>
    </si>
  </sst>
```

In the above example you can see that this time, the text "United States" has specific, colored, formatting applied to the text, "States." *end example*]

18.4.1 charset (Character Set)

This element defines the font character set of this font.

This field is used in font creation and selection if a font of the given facename is not available on the system. Although it is not required to have around when resolving font facename, the information can be stored for when needed to help resolve which font face to use of all available fonts on a system.

Charset represents the basic set of characters associated with a font (that it can display), and roughly corresponds to the ANSI codepage (8-bit or DBCS) of that character set used by a given language. Given more common use of Unicode where many fonts support more than one of the traditional charset categories, and the use of font linking, using charset to resolve font name is less and less common, but still can be useful.

These are operating-system-dependent values.

[*Note*: The following are some of the possible the character sets:

INT Value	Character Set
0	ANSI_CHARSET

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

INT Value	Character Set
1	DEFAULT_CHARSET
2	SYMBOL_CHARSET
77	MAC_CHARSET
128	SHIFTJIS_CHARSET
129	HANGUL_CHARSET
130	JOHAB_CHARSET
134	GB2312_CHARSET
136	CHINESEBIG5_CHARSET
161	GREEK_CHARSET
162	TURKISH_CHARSET
163	VIETNAMESE_CHARSET
177	HEBREW_CHARSET
178	ARABIC_CHARSET
186	BALTIC_CHARSET
204	RUSSIAN_CHARSET
222	THAI_CHARSET
238	EASTEUROPE_CHARSET
255	OEM_CHARSET

The OEM_CHARSET value specifies a character set that is operating-system dependent. *end note*]

Fonts with other character sets can exist in the operating system. If an application uses a font with an unknown character set, it should not attempt to translate or interpret strings that are rendered with that font.

Attributes	Description
val (Value)	The value of an integer, where each value corresponds to a different character set. This attribute is restricted to values ranging from 0 to 255. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_IntProperty) is located in §A.2. *end note*]

18.4.2 outline (Outline)

This element displays only the inner and outer borders of each character. This is very similar to Bold in behavior.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.4.3 phoneticPr (Phonetic Properties)

This element represents a collection of phonetic properties that affect the display of phonetic text for this String Item (si).

Phonetic text is used to give hints as to the pronunciation of an East Asian language, and the hints are displayed as text within the spreadsheet cells across the top portion of the cell. Since the phonetic hints are text, every phonetic hint is expressed as a phonetic run (rPh), and these properties specify how to display that phonetic run.

[*Example*:

```
<si>
  <t>課く 毛こ</t>
  <rPh sb="0" eb="1">
    <t>カ</t>
  </rPh>
  <rPh sb="4" eb="5">
    <t>ケ</t>
  </rPh>
  <phoneticPr fontId="1"/>
</si>
```

The above example shows a String Item that displays some Japanese text "課きく　毛こ." It also displays some phonetic text across the top of the cell. The phonetic text character, "カ" is displayed over the "課" character and the phonetic text "ケ" is displayed above the "毛" character, using the font record in the style sheet at index 1. *end example*]

Attributes	Description
alignment (Alignment)	Specifies how the text for the phonetic run is aligned across the top of the cells, with respect to the main text in the body of the cell. The possible values for this attribute are defined by the ST_PhoneticAlignment simple

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	type (§18.18.56).
fontId (Font Id)	An integer that is a zero-based index into the font record in the style sheet. Represents the font to be used to display this phonetic run. If this index is out of bounds, then the default font of the Normal style should be used in its place. This default font should be at index 0. The possible values for this attribute are defined by the ST_FontId simple type (§18.18.32).
type (Character Type)	An enumeration which specifies which East Asian character set should be used to display the phonetic run The possible values for this attribute are defined by the ST_PhoneticType simple type (§18.18.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PhoneticPr) is located in §A.2. *end note*]

18.4.4 r (Rich Text Run)

This element represents a run of rich text. A rich text run is a region of text that share a common set of properties, such as formatting properties. The properties are defined in the rPr element, and the text displayed to the user is defined in the Text (t) element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RElt) is located in §A.2. *end note*]

18.4.5 rFont (Font)

This element is a string representing the name of the font assigned to display this run.

Attributes	Description
val (String Value)	A string representing the name of the font. If the font doesn't exist (because it isn't installed on the system), or the charset not supported by that font, then another font should be substituted. The string length for this attribute shall be 0 to 31 characters. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontName) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.4.6 rPh (Phonetic Run)

This element represents a run of text which displays a phonetic hint for this String Item (si).

Phonetic hints are used to give information about the pronunciation of an East Asian language. The hints are displayed as text within the spreadsheet cells across the top portion of the cell.

Attributes	Description
eb (Base Text End Index)	An integer used as a zero-based index representing the ending offset into the base text for this phonetic run. This represents the ending point in the base text the phonetic hint applies to. This value shall be between 0 and the total length of the base text. The following condition shall be true: `sb < eb`. It is recommended that the following condition also be satisfied: That for any two consecutive phonetic runs, `sb1 < eb1 <= sb2 < eb2` to avoid overlapping phonetic runs The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sb (Base Text Start Index)	An integer used as a zero-based index representing the starting offset into the base text for this phonetic run. This represents the starting point in the base text the phonetic hint applies to. This value shall be between 0 and the total length of the base text. The following condition shall be true: `sb < eb`. It is recommended that the following condition also be satisfied: That for any two consecutive phonetic runs, `sb1 < eb1 <= sb2 < eb2` to avoid overlapping phonetic runs. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PhoneticRun) is located in §A.2. *end note*]

18.4.7 rPr (Run Properties)

This element represents a set of properties to apply to the contents of this rich text run.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RPrElt) is located in §A.2. *end note*]

18.4.8 si (String Item)

This element is the representation of an individual string in the Shared String table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If the string is just a simple string with formatting applied at the cell level, then the String Item (si) should contain a single text element used to express the string. However, if the string in the cell is more complex - i.e., has formatting applied at the character level - then the string item shall consist of multiple rich text runs which collectively are used to express the string.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rst) is located in §A.2. *end note*]

18.4.9 sst (Shared String Table)

This element is the root of the Shared String Table, which serves as a collection of individual String Items (si).

Attributes	Description
count (String Count)	An integer representing the total count of strings in the workbook. This count does not include any numbers, it counts only the total of text strings in the workbook. This attribute is optional unless `uniqueCount` is used, in which case it is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
uniqueCount (Unique String Count)	An integer representing the total count of unique strings in the Shared String Table. A string is unique even if it is a copy of another string, but has different formatting applied at the character level. [*Example*: World, *World*, and *W*orld. The `count` would be 3, and the `uniqueCount` would be 2. Only one entry for "World" would show in the table because it is the same string, just with different formatting applied at the cell level (i.e., applied to the entire string in the cell). The "*W*orld" string would get a separate unique entry in the shared string table because it has different formatting applied to specific characters. *end example*] This attribute is optional unless `count` is used, in which case it is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sst) is located in §A.2. *end note*]

18.4.10 strike (Strike Through)

This element draws a strikethrough line through the horizontal middle of the text.

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.4.11 sz (Font Size)

This element represents the point size (1/72 of an inch) of the Latin and East Asian text.

Attributes	Description
val (Value)	A double representing the value of a positive measurement in points (1/72 of an inch). The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontSize) is located in §A.2. *end note*]

18.4.12 t (Text)

This element represents the text content shown as part of a string.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.4.13 u (Underline)

This element represents the underline formatting style.

Attributes	Description
val (Underline Value)	An enumeration representing the style of underlining that is used. The none style is equivalent to not using underlining at all. The possible values for this attribute are defined by the ST_UnderlineValues simple type (§18.18.85).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnderlineProperty) is located in §A.2. *end note*]

18.4.14 vertAlign (Vertical Alignment)

This element adjusts the vertical position of the text relative to the text's default appearance for this run. It is used to get 'superscript' or 'subscript' texts, and shall reduce the font size (if a smaller size is available) accordingly.

Attributes	Description
val (Value)	An enumeration representing the vertical-alignment setting. Setting this to either `subscript` or `superscript` shall make the font size smaller if a smaller font size is available. The possible values for this attribute are defined by the ST_VerticalAlignRun simple type (§22.9.2.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VerticalAlignFontProperty) is located in §A.2. *end note*]

18.5 Tables

A table helps organize and provide structure to lists of information in a worksheet. Tables have clearly labeled columns, rows, and data regions. Tables make it easier for users to sort, analyze, format, manage, add, and delete information.

If a region of data is designated as a Table, then special behaviors can be applied which help the user perform useful actions. [*Example*: if the user types additional data in the row adjacent to the bottom of the table, the table can expand and automatically add that data to the data region of the table. Similarly, adding a column is as easy as typing a new column heading to the right or left of the current column headings. Filter and sort abilities can automatically be surfaced to the user via the drop down arrows. Special calculated columns can be created

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

which summarize or calculate data in the table. These columns have the ability to expand and shrink according to size of the table, and maintain proper formula referencing. *end example*]

Tables can be created from data already present in the worksheet, from an external data query, or from mapping a collection of repeating XML elements to a worksheet range.

The sheet XML stores the numeric and textual data. The table XML records the various attributes for the particular table object.

[*Example*:

```
<table xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main"
  id="1" name="MarginTable" displayName="MarginTable" ref="D3:G6"
  totalsRowShown="0">
  <autoFilter ref="D3:G6"/>
  <tableColumns count="4">
    <tableColumn id="1" name="Product"/>
    <tableColumn id="2" name="Wholesale"/>
    <tableColumn id="3" name="Retail"/>
    <tableColumn id="4" name="Margin" dataDxfId="0">
      <calculatedColumnFormula d="1">[Retail]-
        [Wholesale]</calculatedColumnFormula>
    </tableColumn>
  </tableColumns>
  <tableStyleInfo name="TableStyleMedium9" showFirstColumn="0"
    showLastColumn="0" showRowStripes="1" showColumnStripes="0"/>
</table>
```

The above xml example shows a table that spans cells D3 through G6, and has four columns: Product, Wholesale, Retail, and Margin. Margin is a column where each cell has its values calculated based on the formula (Retail - Wholesale), where those values are taken from the cells in the table columns on the corresponding row. The table has a style applied, "TableStyleMedium9", but the styles formatting isn't applied to the first column and the column striping isn't shown. Note that all the data and text values are stored in the sheet xml; the table xml just stores the properties that are specific to this table, and it is referenced by the sheet. *end example*]

18.5.1 Tables

Tables are ranges of data in the worksheet that have special behavior applied which allow users to better sort, analyze, format, manage, add, and delete data. Tables and table columns can also be referenced through formulas by the spreadsheet application using friendly names, making formula calculations that use tables much easier to understand and maintain. Tables provide a natural way for working with large sets of tabular data.

The tables described in this section are of the multi cell variety, as opposed to single cell tables created from XML mappings.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Each table gets its own xml part, and the relationship between a table part and the sheet is defined in the sheet's _rels directory. The sheet xml also references this id since there can be more than one table on a sheet. The sheet xml contains all the numeric and textual data, and the table xml records properties of the table as well as some formatting rules for data and text displayed in the table cells.

18.5.1.1 calculatedColumnFormula (Calculated Column Formula)

Columns in a table can have cells that are calculated, usually based on values in other cells in the table. This element stores the formula that is used to perform the calculation for each cell in this column.

It shall be understood that formulas which reference columns of this table, shall be calculated using the cells in those columns on the same row of the table as the cell that the formula resides in.

See §18.17 for details on the format required for formulas.

Attributes	Description
array (Array)	A Boolean value that indicates whether this formula is an array style formula. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableFormula) is located in §A.2. *end note*]

18.5.1.2 table (Table)

This element is the root element for a table that is not a single cell XML table.

Attributes	Description
comment (Table Comment)	A string representing a textual comment about the table. [*Note*: This can be used by the spreadsheet application in other UI. *end note*] [*Example*: There can be name UI that is used to organize defined names and function references, if tables are listed in that UI the comment can give more information about the table. *end example*] The maximum length of this string should be 32767 characters. The possible values for this attribute are defined by the ST_Xstring simple type

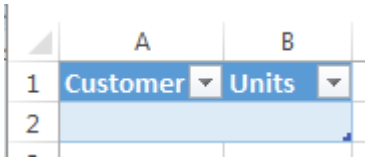
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
connectionId (Connection ID)	An integer representing an ID to indicate which connection from the connections collection is used by this table. This shall only be used for tables that are based off of xml maps. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
dataCellStyle (Data Style Name)	A string representing the name of the cell style that is applied to the data area cells of the table. If this string is missing or does not correspond to the name of a cell style, then the data cell style specified by the current table style should be applied. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dataDxfId (Data Area Format Id)	A zero based integer index into the differential formatting records `<dxfs>` in the styleSheet indicating which format to apply to the data area of this table. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
displayName (Table Name)	A string representing the name of the table. This is the name that shall be used in formula references, and displayed in the UI to the spreadsheet user. This name shall not have any spaces in it, and it shall be unique amongst all other `displayNames` and `definedNames` in the workbook. The character lengths and restrictions are the same as for `definedNames`. See *SpreadsheetML Reference - Workbook definedNames* section for details The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
headerRowBorderDxfId (Header Row Border Format Id)	A zero based integer index into the differential formatting records `<dxfs>` in the styleSheet indicating what border formatting to apply to the header row of this table. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
headerRowCellStyle (Header Row Style)	A string representing the name of the cell style that is applied to the header row cells of the table. If this string is missing or does not correspond to the name of a cell style, then the header

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	row style specified by the current table style should be applied. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
headerRowCount (Header Row Count)	An integer representing the number of header rows showing at the top of the table. 0 means that the header row is not shown. It is up to the spreadsheet application to determine if numbers greater than 1 are allowed. Unless the spreadsheet application has a feature where there might ever be more than one header row, this number should not be higher than 1. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
headerRowDxfId (Header Row Format Id)	A zero based integer index into the differential formatting records <dxfs>in the styleSheet indicating which format to apply to the header row of this table. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
id (Table Id)	A non zero integer representing the unique identifier for this table. Each table in the workbook shall have a unique id. Ids can be used to refer to the specific table in the workbook. [*Note*: For instance a future records bucket could refer to the table using this id. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
insertRow (Insert Row Showing)	A Boolean value indicating whether the insert row should be shown. An insert row indicates to a consuming application that a table has a new row that has no data in it, allowing the application to indicate to the user how to enter data and extend the table. [*Example:* Microsoft Excel formats the insert row using table styling, as shown here:  Although the table contains only headers and covers only row 1, Excel formats row 2 using table styling to indicate that typing in those cells will extend the table to cover them. When the user types into row 2, the insert row is removed. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The insert row should only be shown if the table has no data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
insertRowShift (Insert Row Shift)	A Boolean that indicates whether cells in the sheet had to be inserted when the insert row was shown for this table (see insertRow for details). True if the cells were shifted, false otherwise. [*Note*: This happens when there are values in cells immediately below the table when the table is created and the insert row is shown. In this case blank cells for the insert row are inserted, and the existing values in the sheet are shifted down by one row to make room. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Name)	A string representing the name of the table that is used to reference the table programmatically through the spreadsheet applications object model. This string shall be unique per table per sheet. It has the same length and character restrictions as for displayName. By default this should be the same as the table's `displayName`. This name should also be kept in synch with the `displayName` when the `displayName` is updated in the UI by the spreadsheet user. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
published (Published)	A Boolean representing whether this table is marked as published for viewing by a server based spreadsheet application. True if it should be viewed by the server spreadsheet application, false otherwise. [*Note*: Such an application might only display objects from the workbook that are marked as published, thus being able to load and calculate the entire workbook but only show the specific items that are marked as published. This can allow the server spreadsheet rendering to provide a more restricted view of the workbook. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ref (Reference)	The range on the relevant sheet that the table occupies expressed using A1 style referencing. The reference shall include the totals row if it is shown. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
tableBorderDxfId (Table Border Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating what border formatting to apply to the borders of this table. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
tableType (Table Type)	An optional enumeration specifying the type or source of the table. Indicates whether the table is based off of an external data query, data in a worksheet, or from an xml data mapped to a worksheet. The possible values for this attribute are defined by the ST_TableType simple type (§18.18.78).
totalsRowBorderDxfId (Totals Row Border Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating what border formatting to apply to the totals row of this table. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
totalsRowCellStyle (Totals Row Style)	A string representing the name of the cell style that is applied to the totals row cells of the table. If this string is missing or does not correspond to the name of a cell style, then the totals row style specified by the current table style should be applied. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
totalsRowCount (Totals Row Count)	An integer representing the number of totals rows that shall be shown at the bottom of the table. 0 means that the totals row is not shown. It is up to the spreadsheet application to determine if numbers greater than 1 are allowed. Unless the spreadsheet application has a feature where their might ever be more than one totals row, this number should not be higher than 1. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
totalsRowDxfId (Totals Row Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating which format to apply to the totals row of this table. The spreadsheet should fail to load if this index is out of bounds.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
totalsRowShown (Totals Row Shown)	A Boolean indicating whether the totals row has ever been shown in the past for this table. True if the totals row has been shown, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Table) is located in §A.2. *end note*]

18.5.1.3 tableColumn (Table Column)

An element representing a single column for this table.

Attributes	Description
dataCellStyle (Data Area Style Name)	A string representing the name of the cell style that is applied to the cells in the data area of this table column. If this string is missing or does not correspond to the name of a cell style, then the data cell style specified by the current table style should be applied. This cell style should get precedence over the `dataCellStyle` defined by the `table`. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dataDxfId (Data & Insert Row Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating which format to apply to the data area of this column. This formatting shall also apply to cells on the insert row for this column. See description of attribute insertRow in table (§18.5.1.2) for further information. The spreadsheet should fail to load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
headerRowCellStyle (Header Row Cell Style)	A string representing the name of the cell style that is applied to the header row cell of this column. If this string is missing or does not correspond to the name of a cell style, then header row style specified by the current table style should be applied. This cell style should get precedence over the `headerRowCellStyle` defined by the `table`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
headerRowDxfId (Header Row Cell Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating which format to apply to the header cell of this column. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
id (Table Field Id)	An integer representing the unique identifier of this column. This shall be unique per table. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Column name)	A string representing the unique caption of the table column. This is what shall be displayed in the header row in the UI, and is referenced through functions. This name shall be unique per table. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
queryTableFieldId (Query Table Field Id)	An integer representing the query table field ID corresponding to this table column. The relationship between this table and the corresponding query table is expressed in _rels part for this table. Each queryTableField has a unique `id` attribute, and this `id` is what is referenced here. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
totalsRowCellStyle (Totals Row Style Name)	A string representing the name of the cell style that is applied to the Totals Row cell of this column. If this string is missing or does not correspond to the name of a cell style, then the totals row cell style specified by the current table style should be applied. This cell style should get precedence over the `totalsRowCellStyle` defined by the `table`. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
totalsRowDxfId (Totals Row Format Id)	A zero based integer index into the differential formatting records <dxfs> in the styleSheet indicating which format to apply to the totals row cell of this column. The spreadsheet shall not load if this index is out of bounds. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
totalsRowFunction (Totals Row Function)	An enumeration indicating which type of aggregation to show in the totals row cell for this column. The possible values for this attribute are defined by the ST_TotalsRowFunction simple type (§18.18.83).
totalsRowLabel (Totals Row Label)	A String to show in the totals row cell for this column. This string shall be ignored unless the `totalsRowFunction="none"` for this column, in which case it is displayed in the totals row. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
uniqueName (Unique Name)	An optional string representing the unique name of the table column. This string is used to bind the column to a field in a data table, so it should only be used when this table's `tableType` is `queryTable` or `xml`. This name shall be unique per table when it is used. For tables created from xml mappings, by default this should be the same as the name of the column, and should be kept in synch with the name of the column if that name is altered by the spreadsheet application. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableColumn) is located in §A.2. *end note*]

18.5.1.4 tableColumns (Table Columns)

An element representing the collection of all table columns for this table.

Attributes	Description
count (Column Count)	An integer representing the total count of how many columns there are in this Table. This count shall include both query-defined and user-defined columns. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableColumns) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.5.1.5 tableStyleInfo (Table Style)

This element describes which style is used to display this table, and specifies which portions of the table have the style applied.

Styles define a set of formatting properties that can be easily referenced by cells or other objects in the spreadsheet application. A style can be applied to a table, but tables can define specific parts of the table that should not have the style applied independently of other table parts. For instance a table cannot apply the row striping of the style, and cannot show the style's formatting of the last column, but will apply the column striping and the formatting to the first column.

Attributes	Description
name (Style Name)	A string representing the name of the table style to use with this table. If the style name does not correspond to the name of a table style then the spreadsheet application should use default style. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
showColumnStripes (Show Column Stripes)	A Boolean indicating whether column stripe formatting is applied. True when style column stripe formatting is applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showFirstColumn (Show First Column)	A Boolean indicating whether the first column in the table should have the style applied. True if the first column has the style applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showLastColumn (Show Last Column)	A Boolean indicating whether the last column in the table should have the style applied. True if the last column has the style applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRowStripes (Show Row Stripes)	A Boolean indicating whether row stripe formatting is applied. True when style row stripe formatting is applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleInfo) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.5.1.6 totalsRowFormula (Totals Row Formula)

This element contains a custom formula for aggregating values from the column.

Each tableColumn has a `totalsRowFunction` that can be used for simple aggregations such as average, standard deviation, min, max, count, and others. If a more custom calculation is desired, then this element should be used, and the `totalsRowFunction` shall be set to "`custom`".

Attributes	Description
array (Array)	A Boolean value that indicates whether this formula is an array style formula. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableFormula) is located in §A.2. *end note*]

18.5.1.7 xmlColumnPr (XML Column Properties)

An element defining the XML column properties for a column. This is only used for tables created from XML mappings.

[*Example*: Here is a simple example showing a table column that has an xmlColumPr.

```
<tableColumn id="1" uniqueName="SomeElement" name="SomeElement">
  <xmlColumnPr mapId="1" xpath="/xml/foo/element" xmlDataType="string"/>
</tableColumn>
```

end example]

Attributes	Description

denormalized (Denormalized)	A Boolean that indicates whether the contents of the column have been filled down due to flattening. `True` if it has been filled down (denormilized), `false` otherwise.
	
	This should be used when an XML mapping parent value has many children, and both the parent and child fields are mapped to their own column in the table.
	
	[*Example*:
	`<Order ID="3">`
	`<Item>Milk</Item>`
	`<Item>Bread</Item>`
	`<Item>Cheese</Item>`
	`</Order>`
	
	The resulting table in the spreadsheet application would have two columns, the first with the item ID, filled down for each item in the table as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Item ID Item 3 Milk 3 Bread 3 Cheese *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
mapId (XML Map Id)	An integer representing the ID of the XML map this table field is associated with. The XML map is defined in the xml maps part, and the Map element should have the corresponding `id`. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xmlDataType (XML Data Type)	An enumeration indicating which XML data type is used by this column. The possible values for this attribute are defined by the ST_XmlDataType simple type (§18.18.93).
xpath (XPath)	A string representing the XML path to the element this column is associated with. The spreadsheet application should support XPath limited to the following • The XPath is an absolute path to a simple-content element or attribute • [*Example*: "/ns1:root/ns1:row/ns1:column1" is supported if 'column1' is a child-most node, but not "/ns1:root/ns1:row" for the same document since 'row' is not a child. *end example*] • The XPath does not express axes, but uses the default child axes • [*Example*: "/ns1:root/ns1:row" is supported but not "/ns1:root/child::ns1:row *end example*] • An optional filter can be expressed at the end of the xpath • [*Example*: "/ns1:root/ns1:row/ns1:column1[@foo='abc']" is supported but not "/ns1:root/ns1:row[@foo='abc']/ns1:column1" *end example*] • The filter can only contain a single expression comparing a named attribute to a specific value

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• • Filters are only supported on XPaths that resolve to a simple-content element (not attributes) • • The named attribute shall be defined as an attribute of the simple-content element • • The attribute name shall be preceded by the short-hand (@) symbol representing the axes 'attribute' • [*Example*: "/ns1:root/ns1:row/ns1:column1[@foo='abc']" is supported not "/ns1:root/ns1:row/ns1:column1[attribute::foo='abc']" *end example*] • An arbitrary amount of white-space can be embedded between filter tokens • [*Example*: "/ns1:root/ns1:row/ns1:column1[@ foo='abc']" is permitted*end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_XmlColumnPr) is located in §A.2. *end note*]

18.5.2 Single Cell Tables

A single cell table is generated from an XML mapping. These really just look like regular cells to the spreadsheet user, but shall be implemented as Tables "under the covers."

These tables don't have the full set of properties that multi cell tables do. They only have the various XML properties, and core table properties (such as `id` and `name`) that are needed to create a table and XML mapping. For instance the formatting properties, totals row, and headers row don't exist for the single cell XML tables. The formatting for these cells is maintained in the style sheet.

18.5.2.1 singleXmlCell (Table Properties)

This element represents the table properties for a single cell XML table.

Attributes	Description
connectionId (Connection ID)	An integer representing an ID to indicate which connection from the connections collection is used by this table. This is only used for tables that are based off of xml maps

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Table Id)	An integer representing the unique identifier of the table. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
r (Reference)	An A1 cell style reference to the cell that the single cell xml table occupies. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SingleXmlCell) is located in §A.2. *end note*]

18.5.2.2 singleXmlCells (Single Cells)

This element is a container for a collection of singleXmlCell tables.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SingleXmlCells) is located in §A.2. *end note*]

18.5.2.3 xmlCellPr (Cell Properties)

This element stores the XML properties for the cell of a single cell xml table.

Attributes	Description
id (Table Field Id)	The unique identifier of the XML properties for the cell. This should always be set to the value of 1 since this id is always meant to be for a single cell xml table. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
uniqueName (Unique Table Name)	An optional string representing the unique name of the table column. By default this is the same as the name of the column. This should hold the name of the element or attribute that this cell is referring to in the XML. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_XmlCellPr) is located in §A.2. *end note*]

18.5.2.4 xmlPr (Column XML Properties)

This element represents the column properties for single cell XML tables.

Attributes	Description
mapId (XML Map Id)	An integer representing the ID of the XML map this table field is associated with. The XML map is defined in the xml maps part, and the Map element should have the corresponding `id`. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xmlDataType (XML Data Type)	An enumeration indicating which XML data type is used by this column. The possible values for this attribute are defined by the ST_XmlDataType simple type (§18.18.93).
xpath (XPath)	A string representing the XML path to the element this column is associated with. The spreadsheet application should support XPath limited to the following: • The XPath is an absolute path to a simple-content element or attribute [*Example*: "/ns1:root/ns1:row/ns1:column1" is supported if 'column1' is a child-most node, but not "/ns1:root/ns1:row" for the same document since 'row' is not a child. *end example*] • The XPath does not express axes, but uses the default child axes [*Example*: "/ns1:root/ns1:row" is supported but not "/ns1:root/child::ns1:row *end example*] • An optional filter can be expressed at the end of the xpath [*Example*: "/ns1:root/ns1:row/ns1:column1[@foo='abc']" is supported but not "/ns1:root/ns1:row[@foo='abc']/ns1:column1" *end example*] • The filter can only contain a single expression comparing a named attribute to a specific value • Filters are only supported on XPaths that resolve to a simple-content element (not attributes)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• The named attribute shall be defined as an attribute of the simple-content element • The attribute name shall be preceded by the short-hand (@) symbol representing the axes 'attribute' [*Example*: "/ns1:root/ns1:row/ns1:column1[@foo='abc']" is supported not "/ns1:root/ns1:row/ns1:column1[attribute::foo='abc']" *end example*] • An arbitrary amount of white-space can be embedded between filter tokens [*Example*: "/ns1:root/ns1:row/ns1:column1[@ foo='abc']" is permitted *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_XmlPr) is located in §A.2. *end note*]

18.6 Calculation Chain

The cells in a workbook can be calculated in different orders depending on various optimizations and dependencies. The calculation chain specifies the order in which the cells in a workbook were last calculated.

The calculation chain only deals with cells that require calculation - i.e., it only deals with cells that contain formulas. It does not track or express dependencies amongst the formulas, but rather only records the order in which the cells were last calculated.

The calculation chain order can change over time. One obvious way this can happen is that new formulas can be added, formulas can be removed or updated. The spreadsheet application can also optionally implement partial calculation as an optimization. Partial calculation is when the spreadsheet only recalculates cells that have had their dependencies or values changed. This way, when a number in a cell is changed, requiring an update to a dependent formula, only the cells that are affected by the update is recalculated, as opposed to recalculating the entire workbook.

The calculation chain described in this section is not required by the spreadsheet application, but can be used if the spreadsheet application finds it useful. It can be loaded by a spreadsheet application, or the application can optionally construct it at run time in memory based on formula dependencies. Since the xml data described in this section is not strictly required, the spreadsheet application is free to ignore the order in which the calculation chain specifies calculations - i.e., even if the calculation chain is loaded, the spreadsheet application is free to perform calculations in a different order at run time.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

Consider the following workbook (the formulas shown instead of cell values):

	A	B	C	D	E
1	1	=A1	=B1+A1	=C1+B1+A1	=D1+C1+B1+A1
2					
3					
4					
5	1	=A5	=B5+A5	=C5+B5+A5	=D5+C5+B5+A5
6					
7					

There is a constant entered in A1 and A5, and next to each of those cells are a series of cells which contain formulas that depend on those cells.

After entering the cells on the first row, and then the cells on the 5th row, the calc chain xml looks like this:

```
<calcChain xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main">
  <c r="E5" i="1"/>
  <c r="D5"/>
  <c r="C5"/>
  <c r="B5"/>
  <c r="E1"/>
  <c r="D1"/>
  <c r="C1"/>
  <c r="B1"/>
</calcChain>
```

It is in this order because B1 was calced first (it was the first formula enterd in the workbook), followed by C1, D1, and so on. Then B5 was entered in the 5th row, followed by the other cells in the 5th row, ending with E5.

But, after a full recalculation, the spreadsheet application has realized that cells B5:E5 are on the same child chain, and cells B1:E1 are likewise on their own child chain. The xml now looks like this:

```
<calcChain xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main">
  <c r="B1" i="1"/>
  <c r="C1" s="1"/>
  <c r="D1" s="1"/>
  <c r="E1" s="1"/>
  <c r="B5"/>
  <c r="C5" s="1"/>
  <c r="D5" s="1"/>
  <c r="E5" s="1"/>
</calcChain>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.6.1 c (Cell)

This element represents a single cell, which shall contain a formula, in the calc chain. Cells are calculated in the same order as the c elements appear in the Calculation Chain part.

Attributes	Description
a (Array)	A Boolean flag indicating whether the cell's formula is an array formula. True if this cell's formula is an array formula, false otherwise. If there is a conflict between this attribute and the t attribute of the f element (§18.3.1.40), the t attribute takes precedence. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
i (Sheet Id)	A sheet Id of a sheet the cell belongs to. If this is omitted, it is assumed to be the same as the i value of the previous cell. The possible values for this attribute are defined by the W3C XML Schema int datatype.
l (New Dependency Level)	A Boolean flag indicating that the cell's formula starts a new dependency level. True if the formula starts a new dependency level, false otherwise. Starting a new dependency level means that all concurrent calculations, and child calculations, shall be completed - and the cells have new values - before the calc chain can continue. In other words, this dependency level might depend on levels that came before it, and any later dependency levels might depend on this level; but not later dependency levels can have any calculations started until this dependency level completes. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Cell Reference)	An A-1 style reference to a cell. The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
s (Child Chain)	A Boolean flag indicating whether the cell's formula is on a child chain. True if this cell is part of a child chain, false otherwise. If this is omitted, it is assumed to be the same as the s value of the previous cell . A child chain is a list of calculations that occur which depend on the parent to the chain. There shall not be cross dependencies between child chains. Child chains are not the same as dependency levels - a child chain and its parent are all on the same dependency level. Child chains are series of calculations that can be independently farmed out to other threads or processors. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
t (New Thread)	A Boolean flag indicating whether the cell's formula starts a new thread. True if the cell's formula starts a new thread, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalcCell) is located in §A.2. *end note*]

18.6.2 calcChain (Calculation Chain Info)

This element represents the root of the calculation chain.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalcChain) is located in §A.2. *end note*]

18.7 Comments

A comment is a rich text note that is attached to and associated with a cell, separate from other cell content. Comment content is stored separate from the cell, and is displayed in a drawing object (like a text box) that is separate from, but associated with, a cell. Comments are used as reminders, such as noting how a complex formula works, or to provide feedback to other users. Comments can also be used to explain assumptions made in a formula or to call out something special about the cell.

[*Example*:

```
<comments>
  <authors>
    <author>Bob</author>
    <author>CBR</author>
  </authors>
  <commentList>
    <comment ref="D4" authorId="0">
      <text>
        <r>
          <rPr>
            <b/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <sz val="8"/>
            <color indexed="81"/>
            <rFont val="Calibri"/>
            <charset val="1"/>
            <scheme val="minor"/>
          </rPr>
          <t>Bob:</t>
        </r>
        <r>
          <rPr>
            <sz val="8"/>
           <color indexed="81"/>
           <rFont val="Calibri"/>
           <charset val="1"/>
           <scheme val="minor"/>
         </rPr>
         <t xml:space="preserve">Why such high expense?</t>
       </r>
     </text>
    </comment>
  </commentList>
</comments>
```

end example]

This xml sample displays a comment by "Bob" (bolded) that says, "Why such high expense?" (non bolded).

18.7.1 author (Author)

This element holds a string representing the name of a single author of comments. Every comment shall have an author. The maximum length of the author string is an implementation detail, but a good guideline is 255 chars.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.7.2 authors (Authors)

This element is a container that holds a list of comment author names. There can be many comment authors per sheet, but each author name shall be unique per sheet. The information for each author is stored only once for that sheet, and comments refer to the author by zero based index.

Note that there can be multiple lists of authors per workbook since each sheet contains its own comments part, and each comments part defines a list of authors for comments on that sheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Authors) is located in §A.2. *end note*]

18.7.3 comment (Comment)

This element represents a single user entered comment. Each comment shall have an author and can optionally contain richly formatted text.

Attributes	Description
authorId (Author Id)	Required. An unsigned integer which is used as the zero based index into the list of authors for this set of comments. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
guid (Unique Identifier for Comment)	Unique identifier for this comment. The attribute is required and shall be unique across all comments in shared workbooks. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
ref (Cell Reference)	Required. A string that serves as the A1 style reference to the cell that the comment is associated with. Shall only reference a single cell, not a range of cells, since comments are on a per cell basis. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
shapeId (Shape ID)	Specifies the ID of the DrawingML shape that provides the visual representation of the comment. [*Example*: `<comment shapeId="10 " … >` *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Comment) is located in §A.2. *end note*]

18.7.4 commentList (List of Comments)

This element is a container that holds a list of comments for the sheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommentList) is located in §A.2. *end note*]

18.7.5 commentPr (Comment Properties)

This element specifies the visual and positional properties of a comment.

[*Example*: In the following example, the comment's visual representation resizes with the

```
<comment … >
  <text> … </text>
  <commentPr autoFill="false">
    <anchor moveWithCells="true" sizeWithCells="true">
      <from> … </from>
      <to> … </to>
    <anchor>
  </commentPr>
</comment>
```

end example]

Attributes	Description
altText (Alternative Text)	Specifies alternative text for the object, for use by assistive technologies or applications. [*Example*: `<commentPr altText="Alternate text" … />` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
autoFill (Automatic Fill Flag)	Specifies whether the object's fill formatting is provided automatically by the application. [*Example*: `<commentPr autoFill="false" … />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoLine (Automatic Line Flag)	Specifies whether the object's line formatting is provided automatically by the application. [*Example*: `<commentPr autoLine="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoScale (Automatic Text Scaling Flag)	Specifies whether the object's font is automatically scaled by the application when the object is resized. [*Example*: `<commentPr autoScale="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
colHidden (Hidden Column Flag)	Specifies that the column of the cell to which this comment points is hidden. [*Example*: `<commentPr colHidden="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultSize (Default Size Flag)	Specifies whether the object is at its default size. [*Example*: `<commentPr defaultSize="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
disabled (Disabled Flag)	Specifies whether the object is allowed to run an attached macro. [*Example*: `<commentPr disabled="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
justLastX (Far East Alignment Flag)	Specifies that Far East alignment is set for the last line in the comment's text. Typically, justified text in Far East environments leaves the last line unjustified. Specifying this element also justifies the last line. [*Example*: `<commentPr justLastX="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
locked (Locked Flag)	Specifies that the object is locked when the sheet is protected. [*Example*: `<commentPr locked="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lockText (Text Lock Flag)	Specifies that the object's text is locked. [*Example*: `<commentPr lockText="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
print (Print Flag)	Specifies whether the object is printed when the document is printed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: `<commentPr print="false" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowHidden (Hidden Row Flag)	Specifies that the row of the cell to which this comment points is hidden. [*Example*: `<commentPr rowHidden="true" … />` *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
textHAlign (Text Horizontal Alignment)	Specifies the horizontal alignment of the comment's text field. [*Example*: `<commentPr textHAlign="center" … />` *end example*] The possible values for this attribute are defined by the ST_TextHAlign simple type (§18.18.80).
textVAlign (Text Vertical Alignment)	Specifies the vertical alignment of the comment's text field. [*Example*: `<commentPr textVAlign="center" … />` *end example*] The possible values for this attribute are defined by the ST_TextVAlign simple type (§18.18.81).
uiObject (UI Object Flag)	Specifies whether the object is a UI Object. [*Example*: `<commentPr uiObject="true" … />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommentPr) is located in §A.2. *end note*]

18.7.6 comments (Comments)

This element is the root container of a set of comments and comment authors for a particular sheet. Each set of comments for a sheet is stored in a separate xml part. The relationship part for a sheet defines a link to the correct comment part for that sheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Comments) is located in §A.2. *end note*]

18.7.7 text (Comment Text)

This element contains rich text which represents the text of a comment. The maximum length for this text is a spreadsheet application implementation detail. A recommended guideline is 32767 chars.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rst) is located in §A.2. *end note*]

18.8 Styles

This subclause specifies the possible formatting information for the contents of the cells on a sheet in a SpreadsheetML document.

18.8.1 alignment (Alignment)

Formatting information pertaining to text alignment in cells. There are a variety of choices for how text is aligned both horizontally and vertically, as well as indentation settings, and so on.

Attributes	Description
horizontal (Horizontal Alignment)	Specifies the type of horizontal alignment in cells. The possible values for this attribute are defined by the ST_HorizontalAlignment simple type (§18.18.40).
indent (Indent)	An integer value, where an increment of 1 represents 3 spaces. Indicates the number of spaces (of the normal style font) of indentation for text in a cell. The number of spaces to indent is calculated as following: Number of spaces to indent = indent value * 3

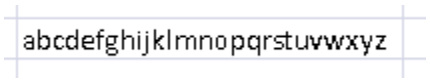
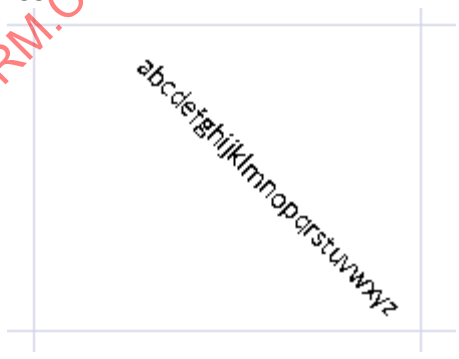
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*:For example, an indent value of 1 means that the text begins 3 space widths (of the normal style font) from the edge of the cell. *end example*] [*Note*: The width of one space character is defined by the font. *end note*] Only `left`, `right`, and `distributed` horizontal alignments are supported. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
justifyLastLine (Justify Last Line)	A boolean value indicating if the cells justified or distributed alignment should be used on the last line of text. (This is typical for East Asian alignments but not typical in other contexts.) The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
readingOrder (Reading Order)	An integer value indicating whether the reading order (bidirectionality) of the cell is left-to-right, right-to-left, or context dependent. 0 - Context Dependent - reading order is determined by scanning the text for the first non-whitespace character: if it is a strong right-to-left character, the reading order is right-to-left; otherwise, the reading order left-to-right. 1 - Left-to-Right- reading order is left-to-right in the cell, as in English. 2 - Right-to-Left - reading order is right-to-left in the cell, as in Hebrew. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
relativeIndent (Relative Indent)	An integer value (used only in a dxf element) to indicate the additional number of spaces of indentation to adjust for text in a cell. The possible values for this attribute are defined by the W3C XML Schema int datatype.
shrinkToFit (Shrink To Fit)	A boolean value indicating if the displayed text in the cell should be shrunk to fit the cell width. Not applicable when a cell contains multiple lines of text. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
textRotation (Text Rotation)	Text rotation in cells. Expressed in degrees. Values are in the range 0 to 180. The first letter of the text is considered the center-point of the arc. For 0 - 90, the value represents degrees above horizon. For 91-180 the degrees below the horizon is calculated as: [degrees below horizon] = 90 - textRotation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	0  45 90 135  180

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
vertical (Vertical Alignment)	Vertical alignment in cells. The possible values for this attribute are defined by the ST_VerticalAlignment simple type (§18.18.88).
wrapText (Wrap Text)	A boolean value indicating if the text in a cell should be line-wrapped within the cell. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellAlignment) is located in §A.2. *end note*]

18.8.2 b (Bold)

Displays characters in bold face font style.

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.8.3 bgColor (Background Color)

Background color of the cell fill pattern. Cell fill patterns operate with two colors: a background color and a foreground color. These combine together to make a patterned cell fill.

Attributes	Description
auto (Automatic)	A boolean value indicating the color is automatic and system color dependent. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indexed (Index)	Indexed color value. Only used for backwards compatibility. References a color in indexedColors. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rgb (Alpha Red Green Blue Color Value)	Standard Alpha Red Green Blue color value (ARGB). The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
theme (Theme Color)	A zero-based index into the `<clrScheme>` collection (§20.1.6.2), referencing a particular `<sysClr>` or `<srgbClr>` value expressed in the Theme part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
tint (Tint)	Specifies the tint value applied to the color. If tint is supplied, then it is applied to the RGB value of the color to determine the final color applied. The tint value is stored as a double from -1.0 .. 1.0, where -1.0 means 100% darken and 1.0 means 100% lighten. Also, 0.0 means no change. In loading the RGB value, it is converted to HLS where HLS values are (0..HLSMAX), where HLSMAX is currently 255. [*Example*: Here are some examples of how to apply tint to color: **If (tint < 0)** Lum’ = Lum * (1.0 + tint) For example: Lum = 200; tint = -0.5; Darken 50% Lum‘ = 200 * (0.5) => 100 For example: Lum = 200; tint = -1.0; Darken 100% (make black)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Lum‘ = 200 * (1.0-1.0) => 0 **If (tint > 0)** Lum‘ = Lum * (1.0-tint) + (HLSMAX – HLSMAX * (1.0-tint)) For example: Lum = 100; tint = 0.75; Lighten 75% Lum‘ = 100 * (1-.75) + (HLSMAX – HLSMAX*(1-.75)) = 100 * .25 + (255 – 255 * .25) = 25 + (255 – 63) = 25 + 192 = 217 For example: Lum = 100; tint = 1.0; Lighten 100% (make white) Lum‘ = 100 * (1-1) + (HLSMAX – HLSMAX*(1-1)) = 100 * 0 + (255 – 255 * 0) = 0 + (255 – 0) = 255 *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Color) is located in §A.2. *end note*]

18.8.4 border (Border)

Expresses a single set of cell border formats (left, right, top, bottom, diagonal). Color is optional. When missing, 'automatic' is implied.

Attributes	Description
diagonalDown (Diagonal Down)	A boolean value indicating if the cell's diagonal border includes a diagonal line, starting at the top left corner of the cell and moving down to the bottom right corner of the cell. [*Example*: This example shows a thin diagonal down line: *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
diagonalUp (Diagonal Up)	A boolean value indicating if the cell's diagonal border includes a diagonal line, starting at the bottom left corner of the cell and moving up to the top right corner of the cell.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: This example shows a thin diagonal up line: *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
outline (Outline)	A boolean value indicating if left, right, top, and bottom borders should be applied only to outside borders of a cell range. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Border) is located in §A.2. *end note*]

18.8.5 borders (Borders)

This element contains borders formatting information, specifying all border definitions for all cells in the workbook.

[*Example*: In this example the first border definition specifies that there are no borders, the second definition specifies that a there is a thin bottom border and medium right border, and the third definition specifies that there is a double top border.

```
<borders count="3">
  <border>
    <begin/>
    <end/>
    <top/>
    <bottom/>
    <diagonal/>
  </border>
  <border>
    <begin/>
    <end style="medium">
      <color indexed="64"/>
    </end>
    <top/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <bottom style="thin">
        <color indexed="64"/>
      </bottom>
      <diagonal/>
    </border>
    <border>
      <begin/>
      <end/>
      <top style="double">
        <color auto="1"/>
      </top>
      <bottom/>
      <diagonal/>
    </border>
  </borders>
```

end example]

Attributes	Description
count (Border Count)	Count of border elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Borders) is located in §A.2. *end note*]

18.8.6 bottom (Bottom Border)

This element specifies the color and line style for the bottom border of a cell.

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.7 cellStyle (Cell Style)

This element represents the name and related formatting records for a named cell style in this workbook.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Annex G contains a listing of cellStyles whose corresponding formatting records are implied rather than explicitly saved in the file. In this case, a builtinId attribute is written on the cellStyle record, but no corresponding formatting records are written.

For all built-in cell styles, the builtinId determines the style, not the name. For all cell styles, Normal is applied by default.

Attributes	Description
builtinId (Built-In Style Id)	The index of a built-in cell style. [*Note*: To maximize interoperability, implementers should restrict the content of this attribute to enumerations present in the list in Annex G.7. Additional values may be used, but interoperability will only be possible via mutual agreement between implementers. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
customBuiltin (Custom Built In)	True indicates that this built-in cell style has been customized. By default built-in styles are not persisted when not in use. This flag indicates that a built-in style has been modified, and therefore should be saved with the workbook, even if not currently in use. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hidden (Hidden Style)	If `'true'` do not show this style in the application UI. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
iLevel (Outline Style)	Indicates that this formatting is for an outline style . When styles are applied to outline levels (using the outline feature), this value is set and the formatting specified on this cell style is applied to the corresponding level of the outline. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (User Defined Cell Style)	The name of the cell style. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
xfId (Format Id)	Zero-based index referencing an xf record in the cellStyleXfs collection. This is used to determine the formatting defined for this named cell style. The possible values for this attribute are defined by the ST_CellStyleXfId simple type (§18.18.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellStyle) is located in §A.2. *end note*]

18.8.8 cellStyles (Cell Styles)

This element contains the named cell styles, consisting of a sequence of named style records. A named cell style is a collection of direct or themed formatting (e.g., cell border, cell fill, and font type/size/style) grouped together into a single named style, and can be applied to a cell.

[*Example*: For example, "Normal", "Heading 1", "Title", and "20% Accent1" are named cell styles expressed below. They have builtInId's associated with them, and use xfId to reference the specific formatting elements pertaining to the particular style. The xfId is a zero-based index, referencing an xf record in the cellStyleXfs collection.

```
<cellStyles count="4">
  <cellStyle name="20% - Accent1" xfId="3" builtinId="30"/>
  <cellStyle name="Heading 1" xfId="2" builtinId="16"/>
  <cellStyle name="Normal" xfId="0" builtinId="0"/>
  <cellStyle name="Title" xfId="1" builtinId="15"/>
</cellStyles>
```

end example]

Attributes	Description
count (Style Count)	Count of style elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellStyles) is located in §A.2. *end note*]

18.8.9 cellStyleXfs (Formatting Records)

This element contains the master formatting records (xf's) which define the formatting for all named cell styles in this workbook. Master formatting records reference individual elements of formatting (e.g., number format, font definitions, cell fills, etc) by specifying a zero-based index into those collections. Master formatting records also specify whether to apply or ignore particular aspects of formatting. [*Example*: Whether to apply a border or not. *end example*]

A cell can have both direct formatting (e.g., bold) and a cell style (e.g., Explanatory) applied to it. Therefore, both the cell style xf records and cell xf records shall be read to understand the full set of formatting applied to a cell.

[*Example*: This example shows 4 master formatting records, each defining formatting for a named cell style (expressed in the cellStyles collection). Note that 0th record does not express any "apply" attributes, while the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

other records do express "apply" attribute values. For example, the last record specifies that number format, alignment, and protection formatting will not be applied to the cell, even when that information is specified in related formatting records.

```
<cellStyleXfs count="4">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0"/>
  <xf numFmtId="0" fontId="2" fillId="0" borderId="0" applyNumberFormat="0"
    applyFill="0" applyBorder="0" applyAlignment="0" applyProtection="0"/>
  <xf numFmtId="0" fontId="3" fillId="0" borderId="1" applyNumberFormat="0"
    applyFill="0" applyAlignment="0" applyProtection="0"/>
  <xf numFmtId="0" fontId="4" fillId="2" borderId="2" applyNumberFormat="0"
    applyAlignment="0" applyProtection="0"/>
</cellStyleXfs>
```

end example]

Attributes	Description
count (Style Count)	Count of cell style (xf) elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellStyleXfs) is located in §A.2. *end note*]

18.8.10 cellXfs (Cell Formats)

This element contains the master formatting records (xf) which define the formatting applied to cells in this workbook. These records are the starting point for determining the formatting for a cell. Cells in the Sheet Part reference the xf records by zero-based index.

A cell can have both direct formatting (e.g., bold) and a cell style (e.g., Explanatory) applied to it. Therefore, both the cell style xf records and cell xf records shall be read to understand the full set of formatting applied to a cell.

Attributes	Description
count (Format Count)	Count of xf elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellXfs) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.8.11 colors (Colors)

Color information associated with this stylesheet. This collection is written whenever the legacy color palette has been modified (backwards compatibility settings) or a custom color has been selected while using this workbook.

When the color palette is modified, the indexedColors collection is written. When a custom color has been selected, the mruColors collection is written.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.2. *end note*]

18.8.12 condense (Condense)

Macintosh compatibility setting. Represents special word/character rendering on Macintosh, when this flag is set. The effect is to condense the text (squeeze it together). SpreadsheetML applications are not required to render according to this flag.

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.8.13 diagonal (Diagonal)

This element specifies the color and line style for the diagonal border(s) of a cell, possibly including diagonally up and diagonally down. The line style for diagonal up and diagonal down lines shall be the same.

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.14 dxf (Formatting)

A single dxf record, expressing incremental formatting to be applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dxf) is located in §A.2. *end note*]

18.8.15 dxfs (Formats)

This element contains the master differential formatting records (dxf's) which define formatting for all non-cell formatting in this workbook. Whereas xf records fully specify a particular aspect of formatting (e.g., cell borders) by referencing those formatting definitions elsewhere in the Styles part, dxf records specify incremental (or differential) aspects of formatting directly inline within the dxf element. The dxf formatting is to be applied on top of or in addition to any formatting already present on the object using the dxf record.

Attributes	Description
count (Format Count)	Count of dxf elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dxfs) is located in §A.2. *end note*]

18.8.16 end (Trailing Edge Border)

This element specifies the color and line style for the trailing edge border of a cell (i.e., the right border for left-to-right cells and the left border for right-to-left cells)..

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.17 extend (Extend)

This element specifies a compatibility setting used for previous spreadsheet applications, resulting in special word/character rendering on those legacy applications, when this flag is set. The effect extends or stretches out the text. SpreadsheetML applications are not required to render according to this flag.

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.8.18 family (Font Family)

The font family this font belongs to. The font name overrides when there are conflicting values.

Attributes	Description
val (Value)	The font family this font belongs to. The possible values for this attribute are defined by the ST_FontFamily simple type (§18.18.94).

[*Note*: The W3C XML Schema definition of this element's content model (ST_FontFamily) is located in §A.2. *end note*]

18.8.19 fgColor (Foreground Color)

Foreground color of the cell fill pattern. Cell fill patterns operate with two colors: a background color and a foreground color. These combine together to make a patterned cell fill.

Attributes	Description
auto (Automatic)	A boolean value indicating the color is automatic and system color dependent. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indexed (Index)	Indexed color value. Only used for backwards compatibility. References a color in indexedColors. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rgb (Alpha Red Green Blue Color Value)	Standard Alpha Red Green Blue color value (ARGB). The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
theme (Theme Color)	A zero-based index into the `<clrScheme>` collection (§20.1.6.2), referencing a particular `<sysClr>` or `<srgbClr>` value expressed in the Theme part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
tint (Tint)	Specifies the tint value applied to the color.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	If tint is supplied, then it is applied to the RGB value of the color to determine the final color applied. The tint value is stored as a double from -1.0 .. 1.0, where -1.0 means 100% darken and 1.0 means 100% lighten. Also, 0.0 means no change. In loading the RGB value, it is converted to HLS where HLS values are (0..HLSMAX), where HLSMAX is currently 255. [*Example*: Here are some examples of how to apply tint to color: **If (tint < 0)** Lum' = Lum * (1.0 + tint) For example: Lum = 200; tint = -0.5; Darken 50% Lum‘ = 200 * (0.5) => 100 For example: Lum = 200; tint = -1.0; Darken 100% (make black) Lum‘ = 200 * (1.0-1.0) => 0 **If (tint > 0)** Lum‘ = Lum * (1.0-tint) + (HLSMAX – HLSMAX * (1.0-tint)) For example: Lum = 100; tint = 0.75; Lighten 75% Lum‘ = 100 * (1-.75) + (HLSMAX – HLSMAX*(1-.75)) = 100 * .25 + (255 – 255 * .25) = 25 + (255 – 63) = 25 + 192 = 217 For example: Lum = 100; tint = 1.0; Lighten 100% (make white) Lum‘ = 100 * (1-1) + (HLSMAX – HLSMAX*(1-1)) = 100 * 0 + (255 – 255 * 0) = 0 + (255 – 0) = 255 *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.2. *end note*]

18.8.20 fill (Fill)

This element specifies fill formatting.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fill) is located in §A.2. *end note*]

18.8.21 fills (Fills)

This element defines the cell fills portion of the Styles part, consisting of a sequence of fill records. A cell fill consists of a background color, foreground color, and pattern to be applied across the cell.

[*Example*: This cell has a yellow fill:

This is the corresponding XML:

```
<fill>
  <patternFill patternType="solid">
    <fgColor rgb="FFFFFF00"/>
    <bgColor indexed="64"/>
  </patternFill>
</fill>
```

This cell has a yellow fill with a thin horizontal crosshatch pattern applied (patternType = `lightGrid`):

This is the corresponding XML:

```
<fill>
  <patternFill patternType="lightGrid">
  <bgColor rgb="FFFFFF00"/>
  </patternFill>
</fill>
```

end example]

Attributes	Description
count (Fill Count)	Count of fill elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fills) is located in §A.2. *end note*]

18.8.22 font (Font)

This element defines the properties for one of the fonts used in this workbook.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Font) is located in §A.2. *end note*]

18.8.23 fonts (Fonts)

This element contains all font definitions for this workbook.

[*Example*: This example expresses two fonts in the workbook. A Calibri family font, with font size of 11, and an Arial family font, with font size 12. The second font has strikethrough applied.

```
<fonts count="2">
  <font>
    <sz val="11"/>
    <color theme="1"/>
    <name val="Calibri"/>
    <family val="2"/>
    <scheme val="minor"/>
  </font>
  <font>
    <strike/>
    <sz val="12"/>
    <color theme="1"/>
    <name val="Arial"/>
    <family val="2"/>
  </font>
</fonts>
```

end example]

Attributes	Description
count (Font Count)	Count of font elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Fonts) is located in §A.2. *end note*]

18.8.24 gradientFill (Gradient)

This element defines a gradient-style cell fill. Gradient cell fills can use one or two colors as the end points of color interpolation.

[*Example*:

This example shows a gradient cell fill, with color green at the top transitioning into blue at the bottom.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This is the XML:

```
<fill>
  <gradientFill degree="90">
    <stop position="0">
      <color rgb="FF92D050"/>
    </stop>
    <stop position="1">
      <color rgb="FF0070C0"/>
    </stop>
  </gradientFill>
</fill>
```

This example shows a gradient cell fill, from the center. Note the left, right, top, and bottom values (and see explanation in the attribute section):

```
<fill>
  <gradientFill type="path" left="0.2" right="0.8" top="0.2" bottom="0.8">
    <stop position="0">
      <color theme="0"/>
    </stop>
    <stop position="1">
      <color theme="4"/>
    </stop>
  </gradientFill>
</fill>
```

end example]

Attributes	Description
bottom (Bottom Convergence)	This attribute is restricted to values ranging from 0 to 1. Specifies in percentage format (from the top to the bottom) the position of the bottom edge of the inner rectangle (color 1). For bottom, 0 means the bottom edge of the inner rectangle is on the top edge of the cell, and 1 means it is on the bottom edge of the cell.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema double datatype.
degree (Linear Gradient Degree)	Angle of the linear gradient - vertical, horizontal, diagonal. [*Example*: In these examples, color 1 is white and color 2 is blue. 90 = Horizontal & color 1 to color 2 270 = Horizontal & color 1 to color 2 0 = Vertical & color 1 to color 2 180 = Vertical & color 1 to color 2 45 = Diagonal Up & top to bottom (color 1 to color 2) 225 = Diagonal Up & bottom to top (color 1 to color 2) 135 = Diagonal Down & top to bottom (color 1 to color 2) 315 = Diagonal Down & bottom to top (color 1 to color 2) *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.
left (Left Convergence)	This attribute is restricted to values ranging from 0 to 1. Specifies in percentage format (from the left to the right) the position of the left edge of the inner rectangle (color 1). For left, 0 means the left edge of the inner rectangle is on the left edge of the cell, and 1 means it is on the right edge of the cell. (applies to From Corner and From Center gradients). The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
right (Right Convergence)	This attribute is restricted to values ranging from 0 to 1. Specifies in percentage format (from the left to the right) the position of the right edge of the inner rectangle (color 1). For right, 0 means the right edge of the inner rectangle is on the left edge of the cell, and 1 means it is on the right edge of the cell. (applies to From Corner and From Center gradients). The possible values for this attribute are defined by the W3C XML Schema double datatype.
top (Top Gradient Convergence)	This attribute is restricted to values ranging from 0 to 1. Specifies in percentage format (from the top to the bottom) the position of the top edge of the inner rectangle (color 1). For top, 0 means the top edge of the inner rectangle is on the top edge of the cell, and 1 means it is on the bottom edge of the cell. (applies to From Corner and From Center gradients). The possible values for this attribute are defined by the W3C XML Schema double datatype.
type (Gradient Fill Type)	Type of this gradient fill. The possible values for this attribute are defined by the ST_GradientType simple type (§18.18.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientFill) is located in §A.2. *end note*]

18.8.25 horizontal (Horizontal Inner Borders)

This element specifies the color and line style for the horizontal inner border(s) of a range of cells. Used in the context of dxf elements only. [*Example*: see the borders definitions for **TableStyleMedium28**. *end example*]

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.26 i (Italic)

Displays characters in italic font style. The italic style is defined by the font at a system level and is not specified by ISO/IEC 29500.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

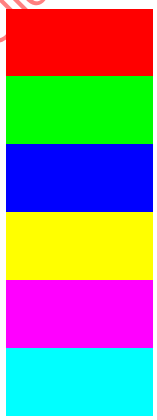
[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.8.27 indexedColors (Color Indexes)

A legacy indexing scheme for colors that is still required for some records, and for backwards compatibility with legacy formats.

This element contains a sequence of RGB color values that correspond to color indexes (zero-based). When using the default indexed color palette, the values are not written out, but instead are implied. When the color palette has been modified from default, then the entire color palette is written out.

Here is the table of default mappings from indexed color value to ARGB value. Note that 0-7 are redundant of 8-15 to preserve backwards compatibility.

Color Index	ARGB Value	[*Example*:
indexed="0"	00000000	
indexed="1"	00FFFFFF	
indexed="2"	00FF0000	
indexed="3"	0000FF00	
indexed="4"	000000FF	
indexed="5"	00FFFF00	
indexed="6"	00FF00FF	
indexed="7"	0000FFFF	
(none)	(none)	
indexed="8"	00000000	
indexed="9"	00FFFFFF	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

indexed="10" 00FF0000

indexed="11" 0000FF00

indexed="12" 000000FF

indexed="13" 00FFFF00

indexed="14" 00FF00FF

indexed="15" 0000FFFF

indexed="16" 00800000

indexed="17" 00008000

indexed="18" 00000080

indexed="19" 00808000

indexed="20" 00800080

indexed="21" 00008080

indexed="22" 00C0C0C0

indexed="23" 00808080

indexed="24" 009999FF

indexed="25" 00993366

indexed="26" 00FFFFCC

indexed="27" 00CCFFFF

indexed="28" 00660066

indexed="29" 00FF8080

indexed="30" 000066CC

indexed="31" 00CCCCFF

indexed="32" 00000080

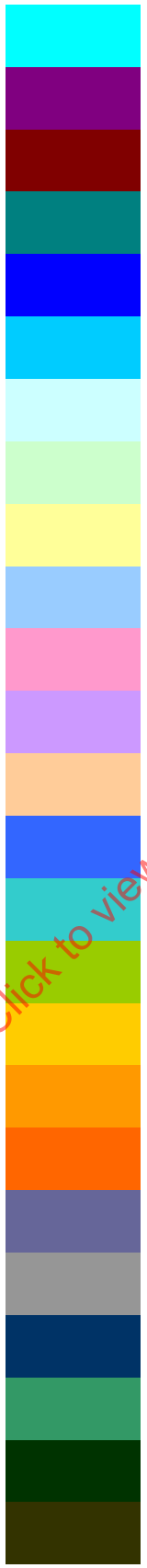
indexed="33" 00FF00FF

indexed="34" 00FFFF00

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

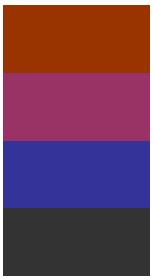
©ISO/IEC 2016 – All rights reserved

indexed="35" 0000FFFF
indexed="36" 00800080
indexed="37" 00800000
indexed="38" 00008080
indexed="39" 000000FF
indexed="40" 0000CCFF
indexed="41" 00CCFFFF
indexed="42" 00CCFFCC
indexed="43" 00FFFF99
indexed="44" 0099CCFF
indexed="45" 00FF99CC
indexed="46" 00CC99FF
indexed="47" 00FFCC99
indexed="48" 003366FF
indexed="49" 0033CCCC
indexed="50" 0099CC00
indexed="51" 00FFCC00
indexed="52" 00FF9900
indexed="53" 00FF6600
indexed="54" 00666699
indexed="55" 00969696
indexed="56" 00003366
indexed="57" 00339966
indexed="58" 00003300
indexed="59" 00333300



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

indexed="60"	00993300	
indexed="61"	00993366	
indexed="62"	00333399	
indexed="63"	00333333	
indexed="64"	System Foreground	n/a
indexed="65"	System Background	n/a

[*Note*: To maximize interoperability, implementers should restrict the content of this attribute to enumerations present in the above list. Additional values may be used, but interoperability will only be possible via mutual agreement between implementers. *end note*]

When values not present in the above list are used, the behavior is implementation-defined.

[*Note*: The W3C XML Schema definition of this element's content model (CT_IndexedColors) is located in §A.2. *end note*]

18.8.28 mruColors (MRU Colors)

This element contains sequence of RGB values that correspond to custom colors selected by the user for this workbook.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MRUColors) is located in §A.2. *end note*]

18.8.29 name (Font Name)

This element specifies the face name of this font.

Attributes	Description
val (String Value)	A string representing the name of the font. If the font doesn't exist (because it isn't installed on the system), or the charset not supported by that font, then another font should be substituted. The string length for this attribute shall be 0 to 31 Unicode scalar values. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontName) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.8.30 numFmt (Number Format)

This element specifies number format properties which indicate how to format and render the numeric value of a cell.

Following is a listing of number formats whose formatCode value is implied rather than explicitly saved in the file. In this case, a numFmtId value is written on the xf record, but no corresponding numFmt element is written. Some of these Ids can be interpreted differently, depending on the UI language of the implementing application.

[*Note*: To maximize interoperability, implementers should restrict the content of this attribute to enumerations present in the lists below. Additional values may be used, but interoperability will only be possible via mutual agreement between implementers. *end note*]

When values not present in the lists below are used, the behavior is implementation-defined.

Ids not specified in the listing, such as 5, 6, 7, and 8, shall follow the number format specified by the formatCode attribute.

All Languages

ID	formatCode
0	General
1	0
2	0.00
3	#,##0
4	#,##0.00
9	0%
10	0.00%
11	0.00E+00
12	# ?/?
13	# ??/??
14	mm-dd-yy
15	d-mmm-yy
16	d-mmm
17	mmm-yy
18	h:mm AM/PM
19	h:mm:ss AM/PM
20	h:mm

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ID	formatCode
21	h:mm:ss
22	m/d/yy h:mm
37	#,##0 ;(#,##0)
38	#,##0 ;[Red](#,##0)
39	#,##0.00;(#,##0.00)
40	#,##0.00;[Red](#,##0.00)
45	mm:ss
46	[h]:mm:ss
47	mmss.0
48	##0.0E+0
49	@

"General" Format

Some additional comments about the "General" number format are appropriate.

The primary goal when a cell is using "General" formatting is to render the cell content without user-specified guidance to the best ability of the application.

Alignment

(Specified for Left-to-Right mode)

- Strings: left aligned
- Boolean/error values: centered
- Numbers: right aligned
- Dates: do not follow the "General" format, instead automatically convert to date formatting.

Numbers

The application shall attempt to display the full number up to 11 digits (inc. decimal point). If the number is too large, the application shall attempt to show exponential format. If the number has too many significant digits, the display shall be truncated. The optimal method of display is based on the available cell width. If the number cannot be displayed using any of these formats in the available width, the application shall show "#" across the width of the cell.

Conditions for switching to exponential format:

1. The cell value shall have at least five digits for xE-xx

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

2. If the exponent is bigger than the size allowed, a floating point number cannot fit, so try exponential notation.
3. Similarly, for negative exponents, check if there is space for even one (non-zero) digit in floating point format.
4. Finally, if there isn't room for all of the significant digits in floating point format (for a negative exponent), exponential format shall display more digits if the exponent is less than -3. (The 3 is because E-xx takes 4 characters, and the leading 0 in floating point takes only 1 character. Thus, for an exponent less than -3, there is more than 3 additional leading 0's, more than enough to compensate for the size of the E-xx.)

Floating point rule:

For general formatting in cells, max overall length for cell display is 11, not including negative sign, but includes leading zeros and decimal separator.

zh-tw and zh-cn

ID	zh-tw formatCode	zh-cn formatCode
27	[$-404]e/m/d	yyyy"年"m"月"
28	[$-404]e"年"m"月"d"日"	m"月"d"日"
29	[$-404]e"年"m"月"d"日"	m"月"d"日"
30	m/d/yy	m-d-yy
31	yyyy"年"m"月"d"日"	yyyy"年"m"月"d"日"
32	hh"時"mm"分"	h"时"mm"分"
33	hh"時"mm"分"ss"秒"	h"时"mm"分"ss"秒"
34	上午/下午 hh"時"mm"分"	上午/下午 h"时"mm"分"
35	上午/下午 hh"時"mm"分"ss"秒"	上午/下午 h"时"mm"分"ss"秒"
36	[$-404]e/m/d	yyyy"年"m"月"
50	[$-404]e/m/d	yyyy"年"m"月"
51	[$-404]e"年"m"月"d"日"	m"月"d"日"
52	上午/下午 hh"時"mm"分"	yyyy"年"m"月"
53	上午/下午 hh"時"mm"分"ss"秒"	m"月"d"日"
54	[$-404]e"年"m"月"d"日"	m"月"d"日"
55	上午/下午 hh"時"mm"分"	上午/下午 h"时"mm"分"
56	上午/下午 hh"時"mm"分"ss"秒"	上午/下午 h"时"mm"分"ss"秒"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ID	zh-tw formatCode	zh-cn formatCode
57	[$-404]e/m/d	yyyy"年"m"月"
58	[$-404]e"年"m"月"d"日"	m"月"d"日"

zh-tw and zh-cn (with unicode values provided for language glyphs where they occur)

ID	zh-tw formatCode	zh-cn formatCode
27	[$-404]e/m/d	yyyy"5E74"m"6708"
28	[$-404]e"5E74"m"6708"d"65E5"	m"6708"d"65E5"
29	[$-404]e"5E74"m"6708"d"65E5"	m"6708"d"65E5"
30	m/d/yy	m-d-yy
31	yyyy"5E74"m"6708"d"65E5"	yyyy"5E74"m"6708"d"65E5"
32	hh"6642"mm"5206"	h"65F6"mm"5206"
33	hh"6642"mm"5206"ss"79D2"	h"65F6"mm"5206"ss"79D2"
34	4E0A5348/4E0B5348hh"6642"mm"5206"	4E0A5348/4E0B5348h"65F6"mm"5206"
35	4E0A5348/4E0B5348hh"6642"mm"5206"ss"79D2"	4E0A5348/4E0B5348h"65F6"mm"5206"ss"79D2"
36	[$-404]e/m/d	yyyy"5E74"m"6708"
50	[$-404]e/m/d	yyyy"5E74"m"6708"
51	[$-404]e"5E74"m"6708"d"65E5"	m"6708"d"65E5"
52	4E0A5348/4E0B5348hh"6642"mm"5206"	yyyy"5E74"m"6708"
53	4E0A5348/4E0B5348hh"6642"mm"5206"ss"79D2"	m"6708"d"65E5"
54	[$-404]e"5E74"m"6708"d"65E5"	m"6708"d"65E5"
55	4E0A5348/4E0B5348hh"6642"mm"5206"	4E0A5348/4E0B5348h"65F6"mm"5206"
56	4E0A5348/4E0B5348hh"6642"mm"5206"ss"79D2"	4E0A5348/4E0B5348h"65F6"mm"5206"ss"79D2"
57	[$-404]e/m/d	yyyy"5E74"m"6708"
58	[$-404]e"5E74"m"6708"d"65E5"	m"6708"d"65E5"

ja-jp and ko-kr

ID	ja-jp formatCode	ko-kr formatCode
27	[$-411]ge.m.d	yyyy"年" mm"月" dd"日"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ID	ja-jp formatCode	ko-kr formatCode
28	[$-411]ggge"年"m"月"d"日"	mm-dd
29	[$-411]ggge"年"m"月"d"日"	mm-dd
30	m/d/yy	mm-dd-yy
31	yyyy"年"m"月"d"日"	yyyy"년" mm"월" dd"일"
32	h"時"mm"分"	h"시" mm"분"
33	h"時"mm"分"ss"秒"	h"시" mm"분" ss"초"
34	yyyy"年"m"月"	yyyy-mm-dd
35	m"月"d"日"	yyyy-mm-dd
36	[$-411]ge.m.d	yyyy"年" mm"月" dd"日"
50	[$-411]ge.m.d	yyyy"年" mm"月" dd"日"
51	[$-411]ggge"年"m"月"d"日"	mm-dd
52	yyyy"年"m"月"	yyyy-mm-dd
53	m"月"d"日"	yyyy-mm-dd
54	[$-411]ggge"年"m"月"d"日"	mm-dd
55	yyyy"年"m"月"	yyyy-mm-dd
56	m"月"d"日"	yyyy-mm-dd
57	[$-411]ge.m.d	yyyy"年" mm"月" dd"日"
58	[$-411]ggge"年"m"月"d"日"	mm-dd

ja-jp and ko-kr (with unicode values provided for language glyphs where they occur)

ID	ja-jp formatCode	ko-kr formatCode
27	[$-411]ge.m.d	yyyy"5E74" mm"6708" dd"65E5"
28	[$-411]ggge"5E74"m"6708"d"65E5"	mm-dd
29	[$-411]ggge"5E74"m"6708"d"65E5"	mm-dd
30	m/d/yy	mm-dd-yy
31	yyyy"5E74"m"6708"d"65E5"	yyyy"B144" mm"C6D4" dd"C77C"
32	h"6642"mm"5206"	h"C2DC" mm"BD84"
33	h"6642"mm"5206"ss"79D2"	h"C2DC" mm"BD84" ss"CD08"
34	yyyy"5E74"m"6708"	yyyy-mm-dd
35	m"6708"d"65E5"	yyyy-mm-dd
36	[$-411]ge.m.d	yyyy"5E74" mm"6708" dd"65E5"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ID	ja-jp formatCode	ko-kr formatCode
50	[$-411]ge.m.d	yyyy"5E74" mm"6708" dd"65E5"
51	[$-411]ggge"5E74"m"6708"d"65E5"	mm-dd
52	yyyy"5E74"m"6708"	yyyy-mm-dd
53	m"6708"d"65E5"	yyyy-mm-dd
54	[$-411]ggge"5E74"m"6708"d"65E5"	mm-dd
55	yyyy"5E74"m"6708"	yyyy-mm-dd
56	m"6708"d"65E5"	yyyy-mm-dd
57	[$-411]ge.m.d	yyyy"5E74" mm"6708" dd"65E5"
58	[$-411]ggge"5E74"m"6708"d"65E5"	mm-dd

th-th

ID	th-th formatCode
59	t0
60	t0.00
61	t#,##0
62	t#,##0.00
67	t0%
68	t0.00%
69	t# ?/?
70	t# ??/??
71	ว/ด/ปปปป
72	ว-ดดด-ปป
73	ว-ดดด
74	ดดด-ปป
75	ช:นน
76	ช:นน:ทท
77	ว/ด/ปปปป ช:นน
78	นน:ทท
79	[ช]:นน:ทท
80	นน:ทท.0

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ID	th-th formatCode
81	d/m/bb

th-th (with unicode values provided for language glyphs where they occur)

ID	th-th formatCode
59	t0
60	t0.00
61	t#,##0
62	t#,##0.00
67	t0%
68	t0.00%
69	t# ?/?
70	t# ??/??
71	0E27/0E14/0E1B0E1B0E1B0E1B
72	0E27-0E140E140E14-0E1B0E1B
73	0E27-0E140E140E14
74	0E140E140E14-0E1B0E1B
75	0E0A:0E190E19
76	0E0A:0E190E19:0E170E17
77	0E27/0E14/0E1B0E1B0E1B0E1B 0E0A:0E190E19
78	0E190E19:0E170E17
79	[0E0A]:0E190E19:0E170E17
80	0E190E19:0E170E17.0
81	d/m/bb

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
formatCode (Number Format Code)	The number format code for this number format. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
numFmtId (Number Format Id)	Id used by the master style records (xf's) to reference this number format. The possible values for this attribute are defined by the ST_NumFmtId simple type

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§18.18.47).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmt) is located in §A.2. *end note*]

18.8.31 numFmts (Number Formats)

This element defines the number formats in this workbook, consisting of a sequence of numFmt records, where each numFmt record defines a particular number format, indicating how to format and render the numeric value of a cell.

[*Example*:

This cell is formatting as US currency:

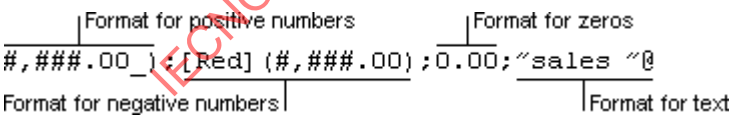
The XML expressing this format shows that the formatId is "166" and the decoded formatCode is `$#,##0.00`

```
<numFmts count="1">
  <numFmt numFmtId="166" formatCode="&quot;$&quot;#,##0.00"/>
</numFmts>
```

end example]

Number Format Codes

Up to four sections of format codes can be specified. The format codes, separated by semicolons, define the formats for positive numbers, negative numbers, zero values, and text, in that order. If only two sections are specified, the first is used for positive numbers and zeros, and the second is used for negative numbers. If only one section is specified, it is used for all numbers. To skip a section, the ending semicolon for that section shall be written.



The first section, "Format for positive numbers", is the format code that applies to the cell when the cell value contains a positive number.

The second section, "Format for negative numbers", is the format code that applies to the cell when the cell value contains a negative number.

The third section, "Format for zeros", is the format code that applies to the cell when the cell value is zero.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The fourth, and last, section, "Format for text", is the format code that applies to the cell when the cell value is text.

The & (ampersand) text operator is used to join, or concatenate, two values.

The following table describes the different symbols that are available for use in custom number formats.

Format symbol	Description and result
0	Digit placeholder. [*Example*: If the value 8.9 is to be displayed as 8.90, use the format #.00 *end example*]
#	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall not display extra zeros when the number typed has fewer digits on either side of the decimal than there are # symbols in the format. [*Example*: If the custom format is #.##, and 8.9 is in the cell, the number 8.9 is displayed. *end example*]
?	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall put a space for insignificant zeros on either side of the decimal point so that decimal points are aligned in the column. [*Example*: The custom format 0.0? aligns the decimal points for the numbers 8.9 and 88.99 in a column. *end example*]
. (period)	Decimal point.
%	Percentage. If the cell contains a number between 0 and 1, and the custom format 0% is used, the application shall multiply the number by 100 and add the percentage symbol in the cell.
, (comma)	Thousands separator. The application shall separate thousands by commas if the format contains a comma that is enclosed by number signs (#) or by zeros. A comma that follows a placeholder scales the number by one thousand. [*Example*: If the format is #.0,, and the cell value is 12,200,000 then the number 12.2 is displayed. *end example*]
E- E+ e- e+	Scientific format. The application shall display a number to the right of the "E" symbol that corresponds to the number of places that the decimal point was moved. [*Example*: If the format is 0.00E+00, and the value 12,200,000 is in the cell, the number 1.22E+07 is displayed. If the number format is #0.0E+0, then the number 12.2E+6 is displayed. *end example*]
$-+():space	Displays the symbol. If it is desired to display a character that differs from one of these symbols, precede the character with a backslash (\). Alternatively, enclose the character in quotation marks. [*Example*: If the number format is (000), and the value 12 is in the cell, the number (012) is displayed. *end example*]
/	If this symbol is preceded and followed by a number symbol (0, #, and ?), it is interpreted as the fraction format symbol and will display the number in the format of a fraction. Otherwise, it is interpreted as the forward slash character and is displayed as such.
\	Displays the next character in the format. The application shall not display the backslash. [*Example*: If the number format is 0\!, and the value 3 is in the cell, the value 3! is displayed. *end example*]
*	Repeats the next character in the format enough times to fill the column to its current width. There shall not be more than one asterisk in one section of the format. If more

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Format symbol	Description and result
	than one asterisk appears in one section of the format, all but the last asterisk shall be ignored. [*Example*: if the number format is 0*x, and the value 3 is in the cell, the value 3xxxxxx is displayed. The number of x characters that are displayed in the cell varies based on the width of the column. *end example*]
_ (underline)	Skips the width of the next character. This is useful for lining up negative and positive values in different cells of the same column. [*Example*: The number format _(0.0_);(0.0) aligns the numbers 2.3 and -4.5 in the column even though the negative number is enclosed by parentheses. *end example*]
"text"	Displays whatever text is inside the quotation marks. [*Example*: The format 0.00 "dollars" displays 1.23 dollars when the value 1.23 is in the cell. *end example*]
@	Text placeholder. If text is typed in the cell, the text from the cell is placed in the format where the at symbol (@) appears. [*Example*: If the number format is "Bob "@" Smith" (including quotation marks), and the value "John" is in the cell, the value Bob John Smith is displayed. *end example*]

Text and spacing

Display both text and numbers

To display both text and numbers in a cell, enclose the text characters in double quotation marks (" ") or precede a single character with a backslash (\). Single quotation marks shall not be used to denote text. Characters inside double quotes, or immediately following backslash shall never be interpreted as part of the format code lexicon; instead they shall always be treated as literal strings. Remember to include the characters in the appropriate section of the format codes. [*Example*: Use the format $0.00" Surplus";$-0.00" Shortage" to display a positive amount as "$125.74 Surplus" and a negative amount as "$-125.74 Shortage." *end example*]

The following characters are displayed without the use of quotation marks.

$	Dollar sign				-	Minus sign
+	Plus sign				/	Slash mark
(	Left parenthesis				)	Right parenthesis
:	Colon				!	Exclamation point
^	Circumflex accent (caret)				&	Ampersand
'	Apostrophe				~	Tilde
{	Left curly bracket				}	Right curly bracket
<	Less-than sign				>	Greater-than sign
=	Equal sign					Space character

Include a section for text entry

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If included, a text section shall be the last section in the number format. Include an "at" sign (@) in the section, precisely where the cell's text value should be displayed. If the @ character is omitted from the text section, text typed in the cell will not be displayed. To always display specific text characters with the typed text, enclose the additional text in double quotation marks (" "). [*Example*: If "June" is typed into the cell, and the text format is "gross receipts for "@ , then the cell will display "gross receipts for June". *end example*]

If the format does not include a text section, text entered in a cell is not affected by the format code.

Add spaces

To create a space that is the width of a character in a number format, include an underscore, followed by the character. [*Example*: When an underscore is followed with a right parenthesis, such as _), positive numbers line up correctly with negative numbers that are enclosed in parentheses because positive numbers are displayed with a blank space after them exactly the width of the right parenthesis character. *end example*]

Repeat characters

To repeat the next character in the format to fill the column width, include an asterisk (*) in the number format. [*Example*: Use 0*- to include enough dashes after a number to fill the cell, or use *0 before any format to include leading zeros. *end example*]

Decimal places, spaces, colors, and conditions

Include decimal places and significant digits

To format fractions or numbers with decimal points, include the following digit placeholders in a section. If a number has more digits to the right of the decimal point than there are placeholders in the format, the number rounds to as many decimal places as there are placeholders. If there are more digits to the left of the decimal point than there are placeholders, the extra digits are displayed. If the format contains only number signs (#) to the left of the decimal point, numbers less than 1 begin with a decimal point.

(number sign) displays only significant digits and does not display insignificant zeros.

0 (zero) displays insignificant zeros if a number has fewer digits than there are zeros in the format.

? (question mark) adds spaces for insignificant zeros on either side of the decimal point so that decimal points align when they are formatted with a fixed-width font, such as Courier New. ? can also be used for fractions that have varying numbers of digits.

To display	As	Use this code
1234.59	1234.6	####.#
8.9	8.900	#.000
.631	0.6	0.#
12 1234.568	12.0 1234.57	#.0#

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

To display	As	Use this code
44.398 102.65 2.8	44.398 102.65 2.8 (with aligned decimals)	???.???
5.25 5.3	5 1/4 5 3/10 (with aligned fractions)	# ???/???

Display a thousands separator

To display a comma as a thousands separator or to scale a number by a multiple of 1,000, include a comma in the number format.

To display	As	Use this code
12000	12,000	#,###
12000	12	#,
12200000	12.2	0.0,,

Specify colors

To set the text color for a section of the format, use the name of one of the following eight colors in square brackets in the section. The color code shall be the first item in the section.

[Black]	[Blue]	[Cyan]
[Green]	[Magenta]	[Red]
[White]	[Yellow]	

Instead of using the name of the color, the color index can be used, like this [Color3] for Red. Numeric indexes for color are restricted to the range from 1 to 56, which reference by index to the legacy color palette.

[*Note*: the default legacy color palette values are listed in §18.8.27. In the format codes, [Color1] refers to the color associated with indexed="8", or black (by default), [Color2] refers to the color associated with indexed="9", or white (by default), and so on up to [Color56] referring to the color associated with indexed="63". If the color palette has been customized from default values, then the colors associated with these indexes will reflect those customizations.

Specify conditions

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

To set number formats that are applied only if a number meets a specified condition, enclose the condition in square brackets. The condition consists of a comparison operator and a value. Comparison operators include: = Equal to; > Greater than; < Less than; >= Greater than or equal to, <= Less than or equal to, and <> Not equal to. [*Example*: The following format displays numbers that are less than or equal to 100 in a red font and numbers that are greater than 100 in a blue font.

```
[Red][<=100];[Blue][>100]
```

end example]

If the cell value does not meet any of the criteria, then pound signs ("#") are displayed across the width of the cell.

Currency, percentages, and scientific notation

Include currency symbols

To include currency symbols, place the currency symbol in the location it should when displayed.

Display percentages

To display numbers as a percentage of 100 — [*Example*: To display .08 as 8% or 2.8 as 280% *end example*]— include the percent sign (%) in the number format.

Display scientific notations

To display numbers in scientific format, use exponent codes in a section — [*Example*: E-, E+, e-, or e+. *end example*]

If a format contains a zero (0) or number sign (#) to the right of an exponent code, the application displays the number in scientific format and inserts an "E" or "e". The number of zeros or number signs to the right of a code determines the number of digits in the exponent. "E-" or "e-" places a minus sign by negative exponents. "E+" or "e+" places a minus sign by negative exponents and a plus sign by positive exponents.

Dates and times

Display days, months, and years

To display	As	Use this code
Months	1–12	m
Months	01–12	mm
Months	Jan–Dec	mmm

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Months	January–December	mmmm
Months	J–D	mmmmm
Days	1–31	d
Days	01–31	dd
Days	Sun–Sat	ddd
Days	Sunday–Saturday	dddd
Years	00–99	yy
Years	date-base minimum value –9999	yyyy

See §18.17.4.1 for details on possible date systems.

Month versus minutes

If "m" or "mm" code is used immediately after the "h" or "hh" code (for hours) or immediately before the "ss" code (for seconds), the application shall display minutes instead of the month.

Display hours, minutes, and seconds

To display	As	Use this code
Hours	0–23	h
Hours	00–23	hh
Minutes	0–59	m
Minutes	00–59	mm
Seconds	0–59	s
Seconds	00–59	ss
Time	4 AM	h AM/PM
Time	4:36 PM	h:mm AM/PM

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Time	4:36:03 P	h:mm:ss A/P
Time	4:36:03.75	h:mm:ss.00
Elapsed time (hours and minutes)	1:02	[h]:mm
Elapsed time (minutes and seconds)	62:16	[mm]:ss
Elapsed time (seconds and hundredths)	3735.80	[ss].00

Minutes versus month

The "m" or "mm" code shall appear immediately after the "h" or "hh" code or immediately before the "ss" code; otherwise, these will display as the month instead of minutes.

AM and PM

If the format contains AM or PM, the hour is based on the 12-hour clock, where "AM" or "A" indicates times from midnight until noon and "PM" or "P" indicates times from noon until midnight. Otherwise, the hour is based on the 24-hour clock.

Illegal date and time values

Cells formatted with a date or time format and which contain date or time values which do not meet the requirements specified shall show the pound sign ("#") across the width of the cell.

International Considerations

Format Code	Description
r	ja-jp/zh-tw only. When loading in ja-jp locale, code becomes "ee". When loading in zh-tw locale, code becomes "e".
rr	ja-jp/zh-tw only. When loading in ja-jp locale, code becomes "gggee". When loading in zh-tw locale, code becomes "e".
g	When loading in ja-jp locale: Single Roman character emperor reign When loading in zh-tw (Taiwan only) locale: treat same as "gg".
gg	When loading in ja-jp locale: Single Kanji character emperor reign

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Format Code	Description
	When loading in zh-tw locale: Last era short name (since 1911)
ggg	When loading in ja-jp locale: Tow Kanji character emperor reign When loading in zh-tw locale: Last era long name (since 1911)
e	When loading in ja-jp locale: Era year When lading in zh-tw (Taiwan only) locale: Era year since 1912. If preceded by “g”, “gg”, or “ggg” then year of 1912, and years before 1912 are special, otherwise years before 1912 are Gregorian. OTHER locales: becomes "yyyy"
ee	When loading in ja-jp locale: Era year w/ leading zero When loading in zh-tw (Taiwan only) locale: Era year since 1911 OTHER locales: becomes "yy"
b2	Hijri calander
b1	Gregorian calendar
[$USD-409]	Specifies currency and locale/date system/number system information. Syntax is [$<Currency String>-<language info>]. Currency string is a string to use as a currency symbol. Language info is a 32-bit value entered in hexidecimal format. Language info format (byte 3 is most significant byte): Bytes 0,1: 16-bit Language ID (LID). Byte 2: Calendar type. High bit indicates that input is parsed using specified calendar. Byte 3: Number system type. High bit indicates that input is parsed using specified number system. Special language info values: 0xf800: System long date format 0xf400: System time format

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
count (Number Format Count)	Count of number format elements. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmts) is located in §A.2. *end note*]

18.8.32 patternFill (Pattern)

This element is used to specify cell fill information for pattern and solid color cell fills. For solid cell fills (no pattern), fgColor is used. For cell fills with patterns specified, then the cell fill color is specified by the bgColor element.

Attributes	Description
patternType (Pattern Type)	Specifies the fill pattern type (including `solid` and none) Default is none, when missing. The possible values for this attribute are defined by the ST_PatternType simple type (§18.18.55).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PatternFill) is located in §A.2. *end note*]

18.8.33 protection (Protection Properties)

Contains protection properties associated with the cell. Each cell has protection properties that can be set. The cell protection properties do not take effect unless the sheet has been protected.

Attributes	Description
hidden (Hidden Cell)	A boolean value indicating if the cell is hidden. When the cell is hidden and the sheet on which the cell resides is protected, then the cell value is displayed in the cell grid location, but the contents of the cell will not be displayed in the formula bar. This is true for all types of cell content, including formula, text, or numbers. Therefore the cell A4 can contain a formula "=SUM(A1:A3)", but if the cell protection property of A4 is marked as hidden, and the sheet is protected, then the cell should display the calculated result [*Example*: "6" *end example*], but will not display the formula used to calculate the result. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
locked (Cell Locked)	A boolean value indicating if the cell is locked. When cells are marked as "locked" and the sheet is protected, then the options specified in the Sheet Part's sheetProtection element (§18.3.1.85) are prohibited for these cells. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CellProtection) is located in §A.2. *end note*]

18.8.34 rgbColor (RGB Color)

A single ARGB entry for the corresponding color index.

Attributes	Description
rgb (Alpha Red Green Blue)	Color value expressed in Alpha Red Green Blue format (ARGB). The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RgbColor) is located in §A.2. *end note*]

18.8.35 scheme (Scheme)

Defines the font scheme, if any, to which this font belongs. When a font definition is part of a theme definition, then the font is categorized as either a major or minor font scheme component. When a new theme is chosen, every font that is part of a theme definition is updated to use the new major or minor font definition for that theme. Usually major fonts are used for styles like headings, and minor fonts are used for body and paragraph text.

Attributes	Description
val (Font Scheme)	Sets font scheme property. The possible values for this attribute are defined by the ST_FontScheme simple type (§18.18.33).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontScheme) is located in §A.2. *end note*]

18.8.36 shadow (Shadow)

Macintosh compatibility setting. Represents special word/character rendering on Macintosh, when this flag is set. The effect is to render a shadow behind, beneath and to the right of the text. SpreadsheetML applications are not required to render according to this flag.

Attributes	Description
val (Value)	A boolean value for the property specified by the parent XML element. If omitted, the default value is `true`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BooleanProperty) is located in §A.2. *end note*]

18.8.37 start (Leading Edge Border)

This element specifies the color and line style for the leading edge border of a cell (i.e., the left border for left-to-right cells and the right border for right-to-left cells).

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.38 stop (Gradient Stop)

One of a sequence of two or more gradient stops, constituting this gradient fill.

Attributes	Description
position (Gradient Stop Position)	Position information for this gradient stop. Interpreted exactly like gradientFill left, right, bottom, top. The position indicated here indicates the point where the color is pure. Before and and after this position the color can be in transition (or pure, depending on if this is the last stop or not). The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientStop) is located in §A.2. *end note*]

18.8.39 styleSheet (Style Sheet)

This is the root element of the Styles part.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Stylesheet) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.8.40 tableStyle (Table Style)

This element represents a single table style definition that indicates how a spreadsheet application should format and display a table.

Each of the tableStyle elements contains a collection of tableStyleElement elements that define formatting for a particular region of the table.

Annex G contains a listing of table styles whose tableStyleElement elements are implied rather than explicitly saved in the file. In this case, a name attribute is written on the tableStyle record, but no corresponding tableStyleElement elements are written.

All of the built-in, named table styles defined in Annex G shall be supported by applications that implement table styles.

[*Note*: Each of the table styles is made up of a collection of formatting definitions, each of which corresponds to a particular structured region of the table. An application can decide to support these built-in types, and can also decide to define more styles, each with their own definitions. An application can also decide whether the user is allowed to customize or further define additional table styles. *end note*]

Attributes	Description
count (Table Style Count)	Count of table style elements defined for this table style. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Table Style Name)	Name of this table style. The possible values for this attribute are defined by the W3C XML Schema string datatype.
pivot (Pivot Style)	'True' if this table style should be shown as an available pivot table style. Not mutually exclusive with table - both can be `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
table (Table)	True if this table style should be shown as an available table style. Not mutually exclusive with pivot - both can be `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyle) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.8.41 tableStyleElement (Table Style)

This element specifies formatting for one area of a table or PivotTable. Together the sequence of these elements makes up one entire Table style or PivotTable style definition.

The order in which table style element formatting is applied is as follows:

Table Style Element Order

- Whole Table
- First Column Stripe
- Second Column Stripe
- First Row Stripe
- Second Row Stripe
- Last Column
- First Column
- Header Row
- Total Row
- First Header Cell
- Last Header Cell
- First Total Cell
- Last Total Cell

For instance, row stripe formatting 'wins' over column stripe formatting, and both 'win' over whole table formatting.

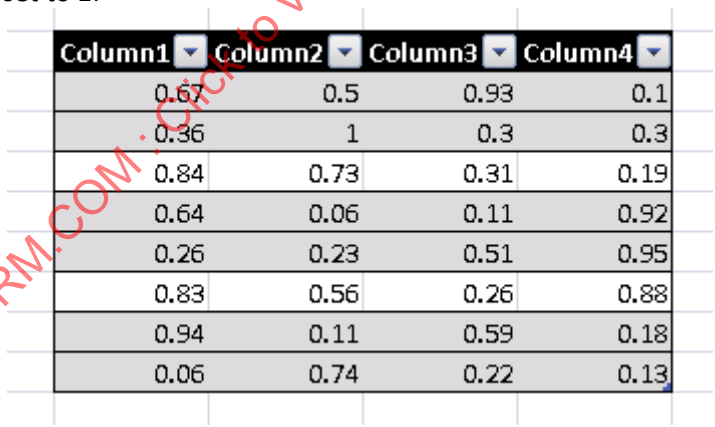
PivotTable Style Element Order

- Whole Table
- Page Field Labels
- Page Field Values
- First Column Stripe
- Second Column Stripe
- First Row Stripe
- Second Row Stripe
- First Column
- Header Row
- First Header Cell
- Subtotal Column 1
- Subtotal Column 2
- Subtotal Column 3
- Blank Row

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Subtotal Row 1
- Subtotal Row 2
- Subtotal Row 3
- Column Subheading 1
- Column Subheading 2
- Column Subheading 3
- Row Subheading 1
- Row Subheading 2
- Row Subheading 3
- Grand Total Column
- Grand Total Row

Attributes	Description
dxfId (Formatting Id)	Zero-based index to a dxf record in the dxfs collection, specifying differential formatting to use with this Table or PivotTable style element. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).
size (Band Size)	Number of rows or columns in a single band of striping. Applies only when type is `firstRowStripe`, `secondRowStripe`, `firstColumnStripe`, or `secondColumnStripe`. [*Example*: In this example, the `firstRowStripe` size is set to `2`, and the `secondRowStripe` size is set to `1`:  *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Table Style Type)	Identifies this table style element's type. The possible values for this attribute are defined by the ST_TableStyleType simple type

Column1	Column2	Column3	Column4
0.67	0.5	0.93	0.1
0.36	1	0.3	0.3
0.84	0.73	0.31	0.19
0.64	0.06	0.11	0.92
0.26	0.23	0.51	0.95
0.83	0.56	0.26	0.88
0.94	0.11	0.59	0.18
0.06	0.74	0.22	0.13

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§18.18.77).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleElement) is located in §A.2. *end note*]

18.8.42 tableStyles (Table Styles)

This element represents a collection of Table style definitions for Table styles and PivotTable styles used in this workbook. It consists of a sequence of tableStyle records, each defining a single Table style.

A Table style is a collection of formatting that applies to structured regions of a Table or PivotTable [*Example*: make the header row & totals bold face, and apply light gray fill to alternating rows in the data portion of the table to achieve striped or banded rows. *end example*]

See the enumeration values in ST_TableStyleType for a listing of structured Table regions to which formatting can be applied, and which together make up a single Table style definition.

Attributes	Description
count (Table Style Count)	Count of table styles defined in this collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
defaultPivotStyle (Default Pivot Style)	Name of the default table style to apply to new PivotTables. This can be set by the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.
defaultTableStyle (Default Table Style)	Name of default table style to apply to new Tables. This can be set by the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyles) is located in §A.2. *end note*]

18.8.43 top (Top Border)

This element specifies the color and line style for the top border of a cell.

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.44 vertical (Vertical Inner Border)

This element specifies the color and line style for the vertical inner border(s) of a range of cells. Used in the context of dxf elements only. [*Example*: see the borders definitions for **TableStyleMedium28**. *end example*]

Attributes	Description
style (Line Style)	The line style for this border. The possible values for this attribute are defined by the ST_BorderStyle simple type (§18.18.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderPr) is located in §A.2. *end note*]

18.8.45 xf (Format)

A single xf element describes all of the formatting for a cell.

Attributes	Description
applyAlignment (Apply Alignment)	A boolean value indicating whether the alignment formatting specified for this xf should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyBorder (Apply Border)	A boolean value indicating whether the border formatting specified for this xf should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyFill (Apply Fill)	A boolean value indicating whether the fill formatting specified for this xf should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyFont (Apply Font)	A boolean value indicating whether the font formatting specified for this xf should be applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyNumberFormat (Apply Number Format)	A boolean value indicating whether the number formatting specified for this xf should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyProtection (Apply Protection)	A boolean value indicating whether the protection formatting specified for this xf should be applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
borderId (Border Id)	Zero-based index of the border record used by this cell format. The possible values for this attribute are defined by the ST_BorderId simple type (§18.18.2).
fillId (Fill Id)	Zero-based index of the fill record used by this cell format. The possible values for this attribute are defined by the ST_FillId simple type (§18.18.30).
fontId (Font Id)	Zero-based index of the font record used by this cell format. The possible values for this attribute are defined by the ST_FontId simple type (§18.18.32).
numFmtId (Number Format Id)	Id of the number format (numFmt) record used by this cell format. The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
pivotButton (Pivot Button)	A boolean value indicating whether the cell should include pivot table sorting and filtering interface elements, in applications intended for editing Office Open XML documents. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
quotePrefix (Quote Prefix)	A boolean value indicating whether the text string in a cell should be prefixed by a single quote mark (e.g., 'text). In these cases, the quote is not stored in the Shared Strings Part. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xfId (Format Id)	For xf records contained in cellXfs this is the zero-based index of an xf record contained in cellStyleXfs corresponding to the cell style applied to the cell. Not present for xf records contained in cellStyleXfs.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_CellStyleXfId simple type (§18.18.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Xf) is located in §A.2. *end note*]

18.9 Metadata

A cell in a spreadsheet application can have metadata associated with it. Metadata is just a set of additional properties about the particular cell, and this metadata is stored in the metadata xml part.

There are two types of metadata: cell metadata and value metadata. Cell metadata contains information about the cell itself, and this metadata can be carried along with the cell as it moves (insert, shift, copy/paste, merge, unmerge, etc). Value metadata is information about the value of a particular cell. Value metadata properties can be propagated along with the value as it is referenced in formulas.

The file format is architected such that it supports both value and cell metadata, as well as even allowing for future extensions. Formulas, such as CUBEMEMBER() or CUBE*, shall make use of value metadata as part of the SpreadsheetML standard. So, only value metadata shall be implemented as it is used by MDX cube functions for retrieving data from OLAP data sources. The other parts are allowed for future extensibility.

See the informative material for background information on OLAP and the various CUBE* functions.

[*Example*: The CUBEMEMBER() function is used to return a specific member from an OLAP cube. The metadata will express the connection name (used as a friendly identifier for the external data connection to the OLAP server and cube), the MDX statement retrieving that member, and a set of operational attributes of the metadata that sepcify how it behaves in the spreadsheet application (i.e., whether it propagates through formula assignment, is able to be copy/pasted, etc).

```
<metadata xmlns="http://purl.oclc.org/ooxml/spreadsheetml/main">
  <metadataTypes count="1">
    <metadataType name="XLMDX" minSupportedVersion="120000" copy="1"
      pasteAll="1" pasteValues="1" merge="1" splitFirst="1" rowColShift="1"
      clearFormats="1" clearComments="1" assign="1" coerce="1"/>
  </metadataTypes>
  <metadataStrings count="2">
    <s v="My Connection"/>
    <s v="[Measures].[Internet Sales Amount]"/>
  </metadataStrings>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <mdxMetadata count="1">
      <mdx n="0" f="m">
        <t c="1">
          <n x="1"/>
        </t>
      </mdx>
    </mdxMetadata>
    <valueMetadata count="1">
      <bk>
        <rc t="1" v="0"/>
      </bk>
    </valueMetadata>
  </metadata>
```

As seen above, the metadata string table contains two entries: the name of the connection (My Connection), and the expression that returns the Internet Sales Amount member from the cube. The metadataType specifies that the metadata persists with assignment, cell merging, copy/pasting, shifting rows/columns, when the formatting or comments are deleted from the cell, and is assigned to the upper left most cell if a merged cell is split. In the valueMetadata collection, the metadata block specifies that the first `metadataType` is used, and indexes the first (0th) entry in the mdxMetadata collection. This MDX element in the mdxMetadata collection in turn specifies the cube function type (m= cube member) and an index into the string table that specifies the connection name. It also contains a tuple (t) element which specifies, via index into the string table, which tuple is returned. *end example*]

[*Note*: When copying a cell with metadata, and the cell contains an array formula, each pasted cell must contain the value from the corresponding position in the array and should contain the metadata corresponding to that cell. *end note*]

18.9.1 bk (Metadata Block)

This element represents a block of metadata records.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataBlock) is located in §A.2. *end note*]

18.9.2 bk (Future Metadata Block)

This element represents a block of future metadata information. This is a location for storing feature extension information.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FutureMetadataBlock) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.9.3 cellMetadata (Cell Metadata)

This element represents cell metadata information. Cell metadata is information metadata about a specific cell, and it stays tied to that cell position.

[*Note*: Applications should not use this for storing metadata, but instead us `valueMetadata`. Cell metadata is included for storing information from future application. *end note*]

Attributes	Description
count (Metadata Block Count)	Number of blocks of metadata records. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataBlocks) is located in §A.2. *end note*]

18.9.4 futureMetadata (Future Metadata)

This element represents future metadata information.

Future data storage areas are xml storage areas that a later version of the spreadsheet application can store data into. So a V2 spreadsheet application can store data for new features that don't exist in the V1 version in a future storage area when saving to a format that the V1 version can open. The V1 version might be able to open the file, but won't necessarily be able to understand data that is stored in a future storage area. So the V1 version might ignore this data, but still round trip it in the file format so that V2 and V1 users can collaborate on the same spreadsheet.

Attributes	Description
count (Future Metadata Block Count)	Number of future metadata blocks. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Metadata Type Name)	Metadata type name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FutureMetadata) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.9.5 k (KPI MDX Metadata)

This element represents key performance indicator (KPI) MDX metadata. A KPI is typically an image that represents the state of some specific business measure at a given point in time. For instance, an image of a green traffic light indicating that customer satisfaction is good.

Attributes	Description
n (Member Unique Name Index)	Index of member unique name in string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
np (KPI Index)	Index of key performance indicator name in string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
p (KPI Property)	Key performance indicator property. The possible values for this attribute are defined by the ST_MdxKPIProperty simple type (§18.18.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MdxKPI) is located in §A.2. *end note*]

18.9.6 mdx (MDX Metadata Record)

This element represents a single record of MDX metadata information which can express a tuple, KPI, set, or member property.

Attributes	Description
f (Cube Function Tag)	This is an enumeration representing the function type of the calling cube function from the spreadsheet. The possible values for this attribute are defined by the ST_MdxFunctionType simple type (§18.18.44).
n (Connection Name Index)	The zero based index of connection name in metadata string store, `metadataStrings`. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Mdx) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.9.7 mdxMetadata (MDX Metadata Information)

This element represents a collection of specific MDX metadata records for the spreadsheet. This is used to build up the members, sets, tuples, KPIs, and member properties for the spreadsheet.

Attributes	Description
count (MDX Metadata Record Count)	Number of MDX metadata metadata records. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MdxMetadata) is located in §A.2. *end note*]

18.9.8 metadata (Metadata)

This element represents the root node for all metadata information in the spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Metadata) is located in §A.2. *end note*]

18.9.9 metadataStrings (Metadata String Store)

This element represents the metadata string store. This is a collection of strings that are used as a resource for the rest of the metadata part. It contains all the required OLAP strings used in the spreadsheet including the connection name, as well as MDX expressions identifying specific members and sets. It is indexed from individual metadata records so that the records can use these strings to build up the necessary MDX statements to retrieve the correct data from the OLAP cube.

Attributes	Description
count (MDX Metadata String Count)	Number of records in the string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataStrings) is located in §A.2. *end note*]

18.9.10 metadataType (Metadata Type Information)

This element represents information about metadata on cells - it defines a specific set of behaviors that the metadata shall adhere to when subject to other spreadsheet operations.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

In general, many of these attributes represent operations that can be performed on a cell that allow the metadata to remain associated with the cell. Operations that are set to 0 or false, will cause the metadata to be disassociated from the cell when that operation is performed.

Attributes	Description
adjust (Adjust Metadata)	A Boolean flag indicating that metadata corresponding to a particular cell needs to be notified when that cell's location is changed. [*Note*: This is included in the file format for future extensibility.*end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
assign (Metadata Formula Assignment)	A Boolean flag indicating whether metadata is propagated by formula assignment operation. True when metadata should be propagated by assignment, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
cellMeta (Cell Metadata)	A Boolean flag indicating whether metadata is cell metadata. True when the metadata is cell metadata, false otherwise - in the false case it is considered to be value metadata. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
clearAll (Metadata Clear All)	A Boolean flag indicating whether metadata survives a "Clear: All" operation. True if the metadata persists after a clear all, false otherwise. The Clear operations can be implemented by the run time application to provide an easy way to allow users to delete everything from a cell (Clear: All), remove only comments (Clear: Comments), only remove formats (Clear: Formats), or only remove the contents but leave the comments and formatting (Clear: Contents). Note these operations can also be performed by the user manually deleting each item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
clearComments (Metadata Clear Comments)	A Boolean flag indicating whether metadata remains after comments have been cleared from the cell. True if the metadata persists after Clear:Comments, false otherwise. The Clear operations can be implemented by the run time application to provide an easy way to allow users to delete everything from a cell (Clear: All), remove only comments (Clear: Comments), only remove formats (Clear: Formats), or only remove the contents but leave the comments and formatting (Clear: Contents). Note these operations can also be performed by the user manually deleting each item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
clearContents	A Boolean flag indicating whether metadata remains after the contents of a cell are

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Metadata Clear Contents)	removed. True if metadata persists after a "Clear: Contents" action, false otherwise. The Clear operations can be implemented by the run time application to provide an easy way to allow users to delete everything from a cell (Clear: All), remove only comments (Clear: Comments), only remove formats (Clear: Formats), or only remove the contents but leave the comments and formatting (Clear: Contents). Note these operations can also be performed by the user manually deleting each item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
clearFormats (Metadata Clear Formats)	A Boolean flag indicating whether metadata remains after formatting is removed from a cell. True if metadata persists after a "Clear: Formats", false otherwise. The Clear operations can be implemented by the run time application to provide an easy way to allow users to delete everything from a cell (Clear: All), remove only comments (Clear: Comments), only remove formats (Clear: Formats), or only remove the contents but leave the comments and formatting (Clear: Contents). Note these operations can also be performed by the user manually deleting each item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
coerce (Metadata Coercion)	A Boolean flag indicating whether value metadata can be removed when this metadata data type is coerced to another data type. True if the value metadata is removed upon coercion, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
copy (Metadata Copy)	A Boolean flag indicating whether metadata is copied with a cell. True if the metadata is copied to other cells when this cell is copied, false otherwise. This shall be set to `true` if the paste attributes for the `metadataType` are going to be used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
delete (Metadata Cell Value Delete)	A Boolean flag indicating whether metadata survives deletion of a cell value. True when the metadata persists after the deletion of a cell value, false otherwise. This attribute is equivalent to the `clearContents` attribute. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
edit (Metadata Edit)	A Boolean flag indicating whether metadata survives the editing of the cell's value. True if the metadata remains unchanged after the cell's value edit, false otherwise.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ghostCol (Metadata Ghost Column)	A Boolean flag indicating whether metadata is copied to/from a ghost column. True when the metadata is copied to/from a ghost column, false otherwise. A ghost column is a single column that exists for the row header. It is not displayed to the end user. It is used to store default formatting for an entire row (i.e. the row gets the formatting for the corresponding cell in the ghost column). For instance, when an entire row is selected and a cell color is applied, this is stored once for the cell in the ghost column instead of for each cell in the row. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ghostRow (Metadata Ghost Row)	A Boolean flag indicating whether metadata is copied to/from a ghost row. True when the metadata is copied to/from a ghost row, false otherwise. A ghost row is a single row that exists for the column header. It is not displayed to the end user. It is used to store default formatting for an entire column (i.e. the column gets the formatting for the corresponding cell in the ghost row). For instance, when an entire column is selected and a cell color is applied, this is stored once for the cell in the ghost row instead of for each cell in the column. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
merge (Metadata Merge)	A Boolean flag indicating whether metadata survives cell merge. True if the metadata persists after a cell merge, false otherwise. It is up to the spreadsheet application on how to deal with conflicts when two cells that each have metadata are merged. The guidance here is to treat it the same as a 'regular' cell merge with the default behavior being that the data in the upper left cell wins. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minSupportedVersion (Minimum Supported Version)	The earliest version of the spreadsheet application that supports this metadata type. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Metadata Type Name)	Represents the name of this particular metadata type. This name shall be unique amongst all other `metadataTypes`. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
pasteAll (Metadata	A Boolean flag indicating whether metadata is populated to a new cell by "Paste: All".

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Paste All)	True when the metadata is populated on a Paste:All, false otherwise. Paste:All and regular paste should be implemented so that they are equivalent by the spreadsheet application. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteBorders (Metadata Paste Borders)	A Boolean flag indicating whether metadata is populated with Paste: Borders. True when the metadata is populated when only borders are pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: The spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteColWidths (Metadata Paste Column Widths)	A Boolean flag indicating whether metadata is populated by Paste: Column Widths. True if the metadata is populated when only column widths are pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteComments (Metadata Paste Comments)	A Boolean flag indicating whether metadata is populated by Paste: Comments. True when metadata is populated when only comments are pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
pasteDataValidation (Metadata Paste Data Validation)	A Boolean flag indicating whether metadata is populated by Paste: Validation. True when metadata is populated when only data validation is pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteFormats (Metadata Paste Formats)	A Boolean flag indicating whether metadata is populated by Paste Special: Formats. True when metadata is populated when only formatting is pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteFormulas (Metadata Paste Formulas)	A Boolean flag indicating whether metadata is populated by Paste: Formulas. True when the metadata is populated when only formulas are pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pasteNumberFormats (Metadata Paste Number Formats)	A Boolean flag indicating whether metadata is populated with Paste: Number Formats. True when metadata is populated when only number formatting is pasted, false otherwise. The copy flag shall be set to `true` for this paste behavior to be respected. [*Note*: the spreadsheet application can implement special pasting behavior, such as pasting everything from a cell (paste all/normal paste), pasting only borders, pasting only comments, or pasting only any other specific cell property. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
pasteValues (Metadata Paste Special Values)	A Boolean flag indicating whether metadata is populated by Paste: Values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowColShift (Metadata Insert Delete)	A Boolean flag indicating whether metadata survives shifting due to row/column insertion/deletion. True if the metadata persists after the cell has been shifted, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
splitAll (Metadata Split All)	A Boolean flag indicating whether a merged cell split action has its metadata copied to all of the resulting cells. True if the metadata is copied to all new cells resulting from a split, false otherwise. If `splitFirst` is also set to `true`, `splitAll` wins - that is all the cells shall have the metadata copied to them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
splitFirst (Meatadata Split First)	A Boolean flag indicating whether when a merged cell is split its metadata is copied to only the first resulting cell. True when the metadata from a split cell is only copied to the first resulting cell, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataType) is located in §A.2. *end note*]

18.9.11 metadataTypes (Metadata Types Collection)

This element is a collection of metadata types.

Attributes	Description
count (Metadata Type Count)	Number of metadata types. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataTypes) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.9.12 ms (Set MDX Metadata)

This element represents an MDX set.

Attributes	Description
c (Sort By Member Index Count)	Number of sort-by member indices. This is essentially the number of coordinates in the cube that this member is defined by. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ns (Set Definition Index)	Zero based index of the set definition in the metadata string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
o (Set Sort Order)	An enumeration specifying what sort order is used to sort the set. The possible values for this attribute are defined by the ST_MdxSetOrder simple type (§18.18.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MdxSet) is located in §A.2. *end note*]

18.9.13 n (Member Unique Name Index)

This element represents an index of a member unique name in metadata string store that is used to define the sort-by set.

Attributes	Description
s (String is a Set)	A Boolean flag indicating whether this string represents a set. True if the string represents a set, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
x (Index Value)	Value of the zero based index. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataStringIndex) is located in §A.2. *end note*]

18.9.14 p (Member Property MDX Metadata)

This element represents an MDX member property.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
n (Member Unique Name Index)	The zero based index of member unique name in the metadata string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
np (Property Name Index)	The zero based index of the property name in metadata string store. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MdxMemeberProp) is located in §A.2. *end note*]

18.9.15 rc (Metadata Record)

This element represents a reference to a specific metadata record.

Attributes	Description
t (Metadata Record Type Index)	A 1-based index to the metadata record type in `metadataTypes`. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
v (Metadata Record Value Index)	A zero based index to a specific metadata record. If the corresponding metadataType has `name`="XLMDX", then this is an index to a record in mdxMetadata, otherwise this is an index to a record in the futureMetadata section whose `name` matches the name of the metadataType. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataRecord) is located in §A.2. *end note*]

18.9.16 t (Tuple MDX Metadata)

This element represents an MDX tuple. A tuple is the intersection of two or more members of distinct dimensions in the cube. For instance, the three members (product, City, month) that are used to show the data point for how many products were sold.

The spreadsheet application should allow the values for the attributes of this element to be specified by the OLAP server.

Attributes	Description
b (Server	A Boolean flag indicating whether the bold style is applied. True if bold shall be applied,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Formatting Bold Font)	false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bc (Server Formatting Background Color)	Specifies the background color in RGB values. It is in hex and is read in the form of 0x00RRGGBB. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
c (Member Index Count)	The number of member expressions in the tuple. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ct (Server Formatting Culture Currency)	The culture tag to use for currency number format. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
fc (Server Formatting Foreground Color)	Represents the foreground color in RGB. It is in hex and is read in the form of 0x00RRGGBB. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
fi (Server Formatting Built-In Number Format Index)	Server formatting built-in number format index. This is an index into the spreadsheet application's built in number formats that is used to specify formatting. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
i (Server Formatting Italic Font)	A Boolean flag indicating that the italic formatting shall be applied. True if italic formatting is applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
si (Server Formatting String Index)	Server formatting string index in the metadata string store, used to index to a string that contains information on how to format the number. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
st (Server Formatting Strikethrough Font)	A Boolean flag indicating whether the strikethrough font style is applied. True if strikethrough shall be applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Server	A Boolean flag indicating whether the underline font style is applied. True if underline

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Formatting Underline Font)	shall be applied, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MdxTuple) is located in §A.2. *end note*]

18.9.17 valueMetadata (Value Metadata)

This element represents the value metadata information for the spreadsheet. It is essentially a collection of block elements that each define the value metadata for a particular cell. Cells in the workbook index into this collection, and each block element in this collection in turn references the `mdxMetadata` records.

Attributes	Description
count (Metadata Block Count)	Number of blocks of metadata records. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MetadataBlocks) is located in §A.2. *end note*]

18.10 Pivot Tables

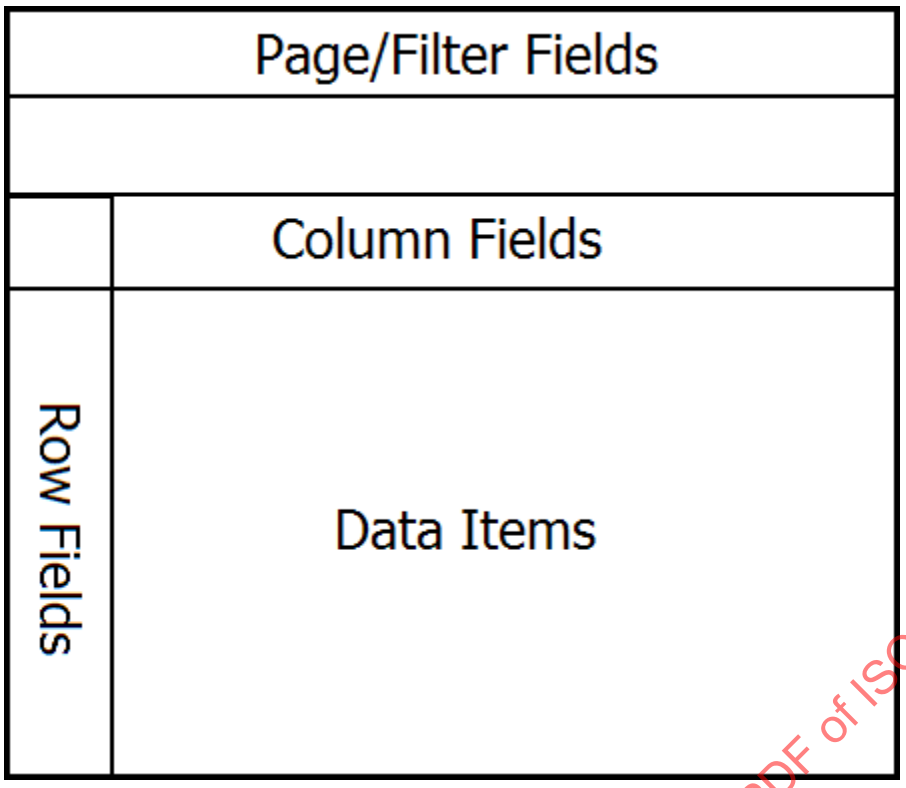
PivotTables display aggregated views of data easily and in an understandable layout. Hundreds or thousands of pieces of underlying information can be aggregated on row and column axes, revealing the meanings behind the data. PivotTable reports are used to organize and summarize your data in different ways. Creating a PivotTable report is about moving pieces of information around to see how they fit together. In a few gestures the pivot rows and columns can be moved into different arrangements and layouts.

A PivotTable object has a row axis area, a column axis area, a data area, and a page/report filter area. Additionally, PivotTables have a corresponding field list pane, or similar user interface, that displays all the fields of data that can be placed on one of the PivotTable areas. In SpreadsheetML, each PivotTable area maps to a collection of fields in the PivotTableDefinition that correspond to each area.

The following image shows the layout for the PivotTable areas.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



[*Example*:

The following image shows a table of data in a worksheet.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	C	F	H	I	O	P	Q	Z	AA	AB
1	Customer Name	Country	City	Product Category	Product Subcategory	Year	Quarter	Month	Sales Amount	Tax Amount	Freight
2	Michele Raman	Australia	Bendigo	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
3	Misty Raji	Australia	Bendigo	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
4	Tabitha E Arthur	Australia	Bendigo	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
5	Clarence D Rai	Australia	Bendigo	Bikes	Mountain Bikes	2001	3	July	3399.99	271.9992	84.9998
6	Jimmy L Moreno	Australia	Bendigo	Bikes	Mountain Bikes	2001	3	July	3399.99	271.9992	84.9998
7	Rob Verhoff	Australia	Bendigo	Bikes	Mountain Bikes	2001	3	July	3374.99	269.9992	84.3748
8	Levi Sai	Australia	Bendigo	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
9	Logan Gonzales	Australia	Brisbane	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
10	Dalton J Lee	Australia	Brisbane	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
11	Jessie J Ortega	Australia	Brisbane	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
12	Paul J. Shakespear	Australia	Caloundra	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
13	Joan R Martin	Australia	Caloundra	Bikes	Road Bikes	2001	3	September	699.0982	55.9279	17.4775
14	Casey Pal	Australia	Caloundra	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
15	Ethan G Coleman	Australia	Caloundra	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
16	Kendra Rubio	Australia	Caloundra	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
17	Bethany G Yuan	Australia	Cloverdal	Bikes	Mountain Bikes	2001	3	August	3399.99	271.9992	84.9998
18	Jasmine Wilson	Australia	Coffs Hart	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
19	Micah Wu	Australia	Coffs Hart	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
20	Warren L Zhang	Australia	Coffs Hart	Bikes	Road Bikes	2001	3	July	699.0982	55.9279	17.4775
21	Ariana Stewart	Australia	Coffs Hart	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
22	Suzanne K Lu	Australia	Coffs Hart	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
23	Randall M Rubio	Australia	Cranbourr	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
24	Deborah K Kumar	Australia	Cranbourr	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
25	Krystal Holt	Australia	Cranbourr	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568
26	Patricia T Raman	Australia	Cranbourr	Bikes	Road Bikes	2001	3	August	3578.27	286.2616	89.4568
27	Wendy Dominguez	Australia	Cranbourr	Bikes	Mountain Bikes	2001	3	August	3374.99	269.9992	84.3748
28	Willie She	Australia	Darlinghu	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
29	Alan Zhu	Australia	Darlinghu	Bikes	Road Bikes	2001	3	September	3578.27	286.2616	89.4568
30	Dawn R Tang	Australia	Darlinghu	Bikes	Road Bikes	2001	3	July	3578.27	286.2616	89.4568

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

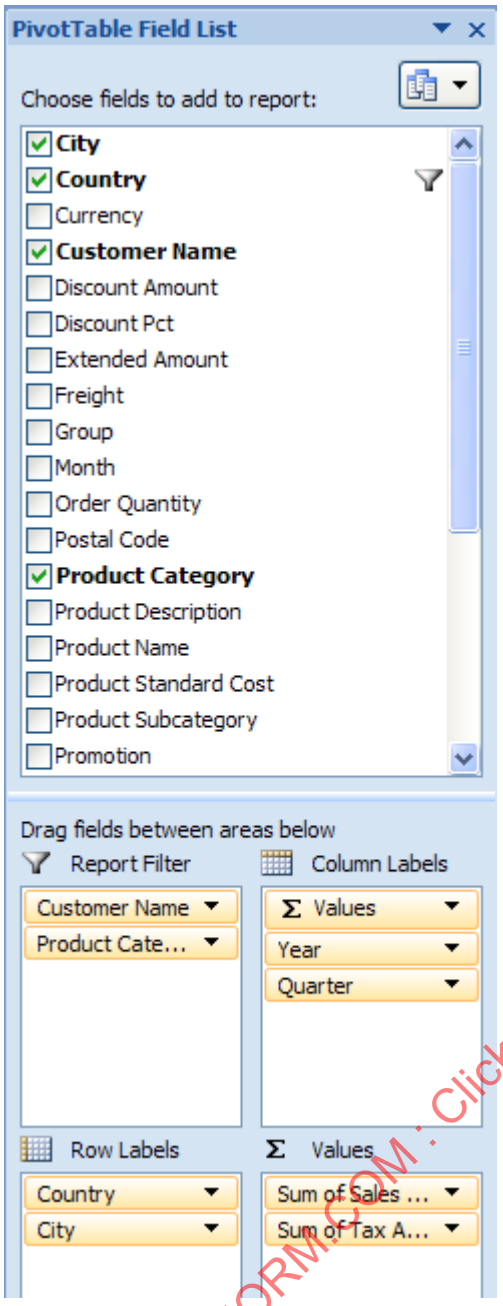
The following image shows a PivotTable summary of the worksheet table data.

	A	B	C	D	E	F	G
1							
2		Country	(All)				
3		State	(All)				
4		City	(All)				
5							
6		Sum of Sales Amount	Column Labels				
7			2001				2001 Total
8			3			3 Total	
9		Row Labels	July	August	September		
10		Bikes	209652.9046	222538.2892	173993.5128	606184.7066	606184.7066
11		Mountain Bikes	64424.81	60899.82	10174.97	135499.6	135499.6
12		Road Bikes	145228.0946	161638.4692	163818.5428	470685.1066	470685.1066
13		Grand Total	209652.9046	222538.2892	173993.5128	606184.7066	606184.7066

The filter area consists of the "Country", "State", and "City" fields. The row area consists of the "Product Category" and "Product Subcategory" fields. "Bikes" belongs to the "Product Category" field and both "Mountain Bikes" and "Road Bikes" belong to the "Product Subcategory" field. The column consists of the "Year" ("2001"), "Quarter" ("3"), and "Month" ("July", "August", and "September") fields.

The following image shows the field list for the PivotTable in the previous image.

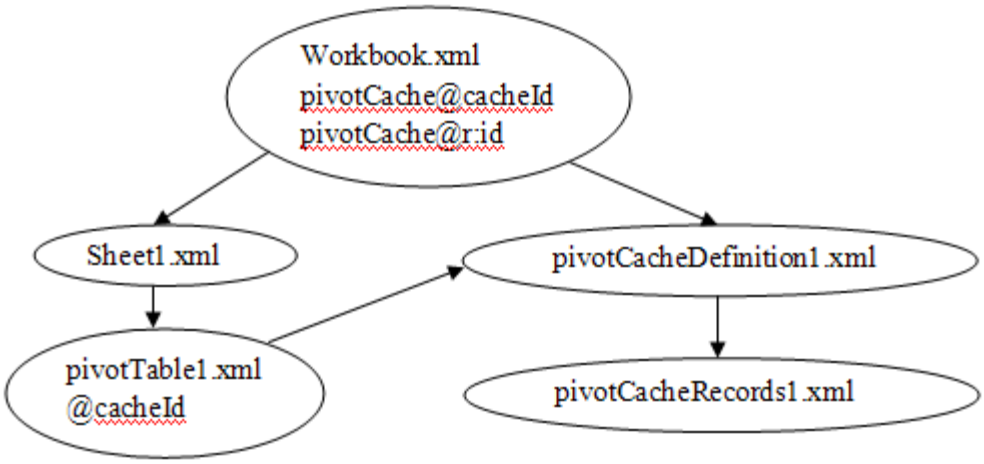
©ISO/IEC 2016 – All rights reserved



File Structure

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



The workbook points to (and owns the longevity of) the *pivotCacheDefinition* part, which in turn points to and owns the *pivotCacheRecords* part. The workbook also points to and owns the sheet part, which in turn points to and owns a *pivotTable* part definition, when a PivotTable is on the sheet. There can be multiple PivotTables on a sheet. The *pivotTable* part points to the appropriate *pivotCacheDefinition* which it is using. Since multiple PivotTables can use the same cache, the *pivotTable* part does not own the longevity of the *pivotCacheDefinition*.

The *pivotTable* part describes the particulars of the layout of the PivotTable on the sheet. It indicates what fields are on the row axis, the column axis, report filter, and values areas of the PivotTable. It also indicates formatting information about the PivotTable. If conditional formatting has been applied to the PivotTable, that is also expressed in the *pivotTable* part.

Outline of XML for pivotTableDefinition

```
<pivotTableDefinition>
  <location/>
  <pivotFields/>
  <rowFields/>
  <rowItems/>
  <colFields/>
  <colItems/>
  <pageFields/>
  <dataFields/>
  <conditionalFormats/>
  <pivotTableStyleInfo/>
</pivotTableDefinition>
```

The *pivotCacheRecords* part contains the underlying data to be aggregated. It is a cache of the source data.

Outline of XML for pivotCacheRecords

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<pivotCacheRecords/>
  <r/>
</pivotCacheRecords>
```

The *pivotCacheDefinition* part defines each field in the *pivotCacheRecords* part, including field name and information about the data contained in the field. The *pivotCacheDefinition* part also defines pivot items that are shared among the *pivotTableDefinition* and *pivotCacheRecords* parts.

Outline of XML for pivotCacheDefinition

```
<pivotCacheDefinition>
  <cacheSource/>
  <cacheFields>
    <cacheField>
      <sharedItems>
        <d/>
      </sharedItems>
      <fieldGroup/>
    </cacheField>
  </cacheFields>
</pivotCacheDefinition>
```

18.10.1 Pivot Tables

This section describes the definition of PivotTables in SpreadsheetML.

18.10.1.1 autoSortScope (AutoSort Scope)

Represents the sorting scope for the PivotTable.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AutoSortScope) is located in §A.2. *end note*]

18.10.1.2 b (Boolean)

Represents a boolean value for an item in the PivotTable.

Attributes	Description
c (Caption)	Specifies the caption for the item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cp (Member Property Count)	Specifies the number of property values for this item. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item)	Specifies a boolean value that indicates whether this item has a calculated value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of 1 or true indicates the item has a calculated value. A value of 0 or false indicates the item does not have a calculated value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of 1 or true indicates this item is unused. A value of 0 or false indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
v (Value)	Specifies the value of the item. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.2. *end note*]

18.10.1.3 cacheField (PivotCache Field)

Represent a single field in the PivotCache. This definition contains information about the field, such as its source, data type, and location within a level or hierarchy. The sharedItems element stores additional information about the data in this field. If there are no shared items, then values are stored directly in the pivotCacheRecords part.

[*Example*:

```
<cacheField name="Group" numFmtId="0">
  <sharedItems count="3">
    <s v="Pacific"/>
    <s v="North America"/>
    <s v="Europe"/>
  </sharedItems>
</cacheField>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
caption (PivotCache Field Caption)	Specifies the caption of the cache field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
databaseField (Database Field)	Specifies a boolean value that indicates whether this field came from the source database rather having been created by the application. A value of `1` or `true` indicates the field is from the source database. A value of `0` or `false` indicates the field was created by the application. [*Note*: This attribute could be used for a defined grouped or calculated field. In this case, source database fields should precede defined grouped or calculated fields. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
formula (Calculated Field Formula)	Specifies the formula for the calculated field. This formula is specified by the end-user. Calculated fields can perform calculations by using the contents of other fields in the PivotTable. In formulas you create for calculated fields or calculated items, you can use operators and expressions as you do in other worksheet formulas. You can use constants and refer to data from the PivotTable, but you cannot use cell references or defined names. You cannot use worksheet functions that require cell references or defined names as arguments, and you cannot use array functions. Further behaviors and restrictions apply to formulas for calculated fields: • Formulas for calculated fields operate on the sum of the underlying data for any fields in the formula. [*Example*: The formula =Sales * 1.2 multiplies the sum of the sales for each type and region by 1.2; it does not multiply each individual sale by 1.2 and then sum the multiplied amounts. *end example*] • Formulas cannot refer to totals. For more information about formulas see §18.17 in Formulas. For more information about defined names see §18.2.6 in Workbook. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
hierarchy (Hierarchy)	Specifies the hierarchy that this field is part of. The possible values for this attribute are defined by the W3C XML Schema int datatype.
level (Hierarchy Level)	Specifies the hierarchy level that this field is part of. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
mappingCount (Member Property Count)	Specifies the number of property mappings for this field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
memberPropertyField (Member Property Field)	Specifies a boolean value that indicates whether the field contains OLAP member property information. A value of `1` or `true` indicates this field contains OLAP member property information. A value of `0` or `false` indicates this field does not contain OLAP member property information. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (PivotCache Field Name)	Specifies the name of the cache field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
numFmtId (Number Format Id)	Specifies the number format that is applied to all items in the field. Number formats are written to the styles part. For more information see §18.8.31 in Styles. [*Note*: Formatting information provided by cell table and by PivotTable need not agree. If the two formats differ, the cell-level formatting takes precedence. If you change the layout of the PivotTable, the PivotTable formatting will then take precedence. *end note*] The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
propertyName (Property Name)	Specifies the name of the property if this field is an OLAP property field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
serverField (Server-based Field)	Specifies a boolean value that indicates whether the field is a server-based page field. A value of `1` or `true` indicates this field is a server-based page field. A value of `0` or `false` indicates this field is not a server-based page field. This attribute applies to ODBC sources only. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sqlType (SQL Data Type)	Specifies the SQL data type of the field. This attribute applies to ODBC data sources only. A value is supplied for this attribute only if it is provided to the application.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• For more information, including supported data types, see ISO 9075-3:2008, Table 33 - Codes used for concise data types. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueList (Unique List Retrieved)	Specifies a boolean value that indicates whether the application was able to get a list of unique items for the field. The attribute only applies to PivotTables that use ODBC and is intended to be used in conjunction with optimization features in the application. [*Example*: the application can optimize memory usage when populating PivotCache records if it has a list of unique items for a field before all the records are retrieved from ODBC. *end example*] A value of `1` or `true` indicates the application was able to get a list of unique values for the field. A value of `0` or `false` indicates the application was unable to get a list of unique values for the field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CacheField) is located in §A.2. *end note*]

18.10.1.4 cacheFields (PivotCache Fields)

Represents the collection of field definitions in the source data.

[*Example*:

```
<cacheFields count="1">
  <cacheField name="Group" numFmtId="0">
    <sharedItems count="3">
      <s v="One"/>
      <s v="Two"/>
      <s v="Three"/>
    </sharedItems>
  </cacheField>
</cacheFields>
```

end example]

Attributes	Description
count (Field Count)	Specifies the number of fields in the cache.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CacheFields) is located in §A.2. *end note*]

18.10.1.5 cacheHierarchies (PivotCache Hierarchies)

Represents the collection of OLAP hierarchies in the PivotCache.

[*Example*:

```
<cacheHierarchies count="2">
  <cacheHierarchy uniqueName="[Account].[Account]" caption="Account"
    attribute="1" keyAttribute="1"
    defaultMemberUniqueName="[Account].[Account].[All Accounts]"
    allUniqueName="[Account].[Account].[All Accounts]"
    dimensionUniqueName="[Account]" count="0"/>
  <cacheHierarchy uniqueName="[Account].[Account Number]" caption="Account
    Number" attribute="1" defaultMemberUniqueName="[Account].[Account
    Number].[All Accounts]" allUniqueName="[Account].[Account Number].[All
    Accounts]" dimensionUniqueName="[Account]" count="0"/>
</cacheHierarchies>
```

end example]

Attributes	Description
count (Hierarchy Count)	Specifies the number of OLAP hierarchies in the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CacheHierarchies) is located in §A.2. *end note*]

18.10.1.6 cacheHierarchy (PivotCache Hierarchy)

Represents an OLAP hierarchy in the PivotCache.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<cacheHierarchy uniqueName="[Account].[Account Number]" caption="Account Number"
  attribute="1" defaultMemberUniqueName="[Account].[Account Number].[All
  Accounts]" allUniqueName="[Account].[Account Number].[All Accounts]"
  dimensionUniqueName="[Account]" count="0"/>
```

end example]

Attributes	Description
allCaption (Display Name of 'All')	Specifies the display name of the "all" member of this hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
allUniqueName (Unique Name of 'All')	Specifies the unique name of the "all" member of this hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
attribute (Attribute Hierarchy)	Specifies a boolean value that indicates whether this hierarchy is an attribute hierarchy. An attribute hierarchy is an OLAP member that is exposed as a flat, single-level hierarchy on the OLAP server. A value of `1` or `true` indicates this hierarchy is an attribute hierarchy. A value of `0` or `false` indicates this hierarchy is not an attribute hierarchy. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
caption (Hierarchy Display Name)	Specifies the display name of the hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
count (Levels Count)	Specifies the number of levels in this hierarchy. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
defaultMemberUniqueName (Default Member Unique Name)	Specifies the unique name of the default member of this hierarchy The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dimensionUniqueName (Dimension Unique Name)	Specifies the unique name of the dimension to which this hierarchy belongs. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
displayFolder (Display Folder)	Specifies the display folder in which this hierarchy should be displayed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
hidden (Hidden)	Specifies a boolean value that indicates whether the hierarchy is hidden. A value of `1` or `true` indicates this hierarchy is hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
iconSet (KPI Icon Set)	Specifies the icon set to use to visualize a KPI trend or status expression. PivotTables use the icon sets available for conditional formatting in SpreadsheetML. See associated simple type definition for details. The following values are used by PivotTables: • no value: default iconset. For status KPI this corresponds to 3 traffic lights. For trend KPI this corresponds to 3-arrows. • 1: Variance Arrow - 3 arrow. • 2: 3 arrows • 3: Status Arrow Ascending - 5 arrows. • 4: Status Arrow Descending - 5 arrows. • 5: Standard Arrow - 5 arrows gray. • 6: Traffic Light Single - 3 traffic lights 1. • 7: Traffic Light, Traffic Light Multiple - 3 traffic lights 2. • 8: Gauge Ascending - 5 quarters. • 9: Gauge Descending - 5 quarters. • 10: Thermometer, Cylinder, Smiley Face - 3 signs. • 11: Road Signs - 3 symbols. The possible values for this attribute are defined by the W3C XML Schema int datatype.
keyAttribute (Key Attribute Hierarchy)	Specifies a boolean value that indicates whether this hierarchy is the key attribute hierarchy in an OLAP dimension. A value of `1` or `true` indicates this hierarchy is the key attribute hierarchy in an OLAP dimension. A value of `0` or `false` indicates this hierarchy is not a key attribute hierarchy. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
measure (Measure Hierarchy)	Specifies a boolean value that indicates whether this hierarchy is a measure. A value of `1` or `true` indicates this hierarchy is a measure. A value of `0` or `false` indicates this hierarchy is not a measure. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
measureGroup (Measure Group Name)	Specifies the name of the measure group to which this hierarchy belongs. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
measures (Measures)	Specifies a boolean value that indicates whether this hierarchy contains all the measures. A value of `1` or `true` indicates this hierarchy contains all the measures. A value of `0` or `false` indicates this hierarchy does not contain all the measures. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
memberValueDatatype (Member Value Data Type)	Specifies the data type of the member value. This attribute stores an OLEDB data type. [*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbtype_indicators.asp *end note*] `memberValueDataType` is stored for key attribute hierarchies in order to tell when the application will offer date filtering instead of label filtering in OLAP PivotTables. Date filtering is only offered when the data type is Date/Time. `memberValueDatatype="7"` indicates a date/time data type. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
oneField (One Field)	Specifies a boolean value that indicates whether this hierarchy is associated with only one field due to its position in the view. A value of `1` or `true` indicates this hierarchy is associated with only one field. A value of `0` or `false` indicates this field is not restricted to only one association due to its position in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
parentSet (Parent Set)	Specifies the parent hierarchy of the set. If the attribute is missing it means that the parent hierarchy is unknown or doesn't exist in the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
set (Set)	Specifies a boolean value that indicates whether this hierarchy is a set. A value of `1` or `true` indicates this hierarchy is a set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates this hierarchy is not a set. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
time (Time)	Specifies a boolean value that indicates whether this hierarchy is of type time. A value of `1` or `true` indicates this hierarchy is of type time. A value of `0` or `false` indicates is of a different type. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
unbalanced (Unbalanced)	Specifies a boolean value that indicates whether this hierarchy is an unbalanced hierarchy. If value is not written, then this attribute either cannot be determined or does not apply to the current hierarchy. A value of `1` or `true` indicates this hierarchy is unbalanced. A value of `0` or `false` indicates is balanced. For more information on balanced hierarchies, see the documentation provided for your OLAP server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
unbalancedGroup (Unbalanced Group)	Specifies a boolean value that indicated whether the grouped version of this hierarchy is an unbalanced hierarchy. If value is not written, then this attribute either cannot be determined or does not apply to the current hierarchy. A value of `1` or `true` indicates this hierarchy is unbalanced when grouped. A value of `0` or `false` indicates is balanced when grouped. For more information on balanced hierarchies, see the documentation provided for your OLAP server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uniqueName (Hierarchy Unique Name)	Specifies the unique name of the hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CacheHierarchy) is located in §A.2. *end note*]

18.10.1.7 cacheSource (PivotCache Source Description)

Represents the description of data source whose data is stored in the pivot cache. The data source refers to the underlying rows or database records that provide the data for a PivotTable. You can create a PivotTable report from a SpreadsheetML table, an external database (including OLAP cubes), multiple SpreadsheetML worksheets, or another PivotTable.

Quarter	Region	Sport	Sales
Qtr1	East	Golf	$5,000
Qtr1	East	Safari	$9,000
Qtr1	East	Tennis	$1,500
Qtr2	East	Golf	$2,000
Qtr2	East	Safari	$6,000
Qtr2	East	Tennis	$500
Qtr1	West	Golf	$3,500
Qtr1	West	Tennis	$6,000
Qtr2	West	Golf	$2,500
Qtr2	West	Tennis	$3,200

Information about the data source is stored in the connection element and is retrieved using the connectionId attribute.

[*Example*:

```
<cacheSource type="external" connectionId="1"/>
```

end example]

OLAP data sources are distinguished from other data sources in SpreadsheetML. OLAP records are not stored in the *pivotCacheRecords* part, whereas all records for non-OLAP data sources are stored in the cache.

Attributes	Description
connectionId (Connection Index)	Specifies the index to the workbook connection. This attribute is used when the cache type is 'External.' See §18.13.1 for more information about the connection element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Cache Type)	Specifies the cache type. The possible values for this attribute are defined by the ST_SourceType simple type (§18.18.75).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CacheSource) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.8 calculatedItem (Calculated Item)

Represents an item within a PivotTable field that uses a formula . The formula is specified in the formula attribute.

Calculations and options available for a PivotTable depend on whether the source data came from an OLAP database or another type of database. This complex type applies to non-OLAP external data or on worksheet data. See calculatedMember for information on calculations on OLAP data sources.

Attributes	Description
field (Field Index)	Specifies the index of the pivotField with which this calculated item is associated. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
formula (Calculated Item Formula)	Specifies the formula of the calculated item. In formulas you create for calculated items, you can use operators and expressions as you do in other worksheet formulas. You can use constants and refer to data from the PivotTable, but you cannot use cell references or defined names. You cannot use worksheet functions that require cell references or defined names as arguments, and you cannot use array functions. Further behaviors and restrictions apply to formulas for `calculatedItems`: • Formulas for calculated items operate on the individual records; the calculated item formula =Dairy *115% multiplies each individual sale of Dairy times 115%, after which the multiplied amounts are summarized together in the data area. • Formulas cannot refer to totals. • You can include the field name in a reference to an item. The item name shall be in square brackets. Use this format to avoid #NAME? errors when two items in two different fields in a report have the same name. • You can refer to an item by its position in the PivotTable as currently sorted and displayed. The item referred to in this way can change whenever the positions of items change or different items are displayed or hidden. Hidden items are not counted in this index. • You can use relative positions to refer to items. The positions are determined relative to the calculated item that contains the formula. If the position you give is before the first item or after the last item in the field, the formula results in a #REF! error. For more information about formulas see §18.17 in Formulas. For more information about defined names see §18.2.6 in Workbook. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalculatedItem) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.10.1.9 calculatedItems (Calculated Items)

Represents the collection of calculated items.

Attributes	Description
count (Calculated Item Formula Count)	Specifies the number of calculated item formulas in the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalculatedItems) is located in §A.2. *end note*]

18.10.1.10 calculatedMember (Calculated Member)

A calculated member is a member in an OLAP hierarchy for which the value is calculated by an OLAP server using a Multidimensional Expressions (MDX) expression. For PivotTables that are created from OLAP cubes the summarized values are calculated by an OLAP server before the SpreadsheetML application displays the results. In OLAP PivotTables, the consuming application cannot change the summary function used to calculate totals and subtotals.
Calculated members are defined by the Multidimensional Expressions (MDX) expression in the mdx attribute.

[*Example*:

```
<calculatedMembers count="1">
  <calculatedMember name="[Product].[Product Categories].[All
    Products].[Calculated Member]" mdx="'[Product].[Product Categories].[All
    Products].[Accessories]'" memberName="Calculated Member"
    hierarchy="[Product].[Product Categories]" parent="[Product].[Product
    Categories].[All Products]"/>
</calculatedMembers>
```

end example]

Attributes	Description
hierarchy (Hierarchy Name)	Specifies the name of the hierarchy to which the calculated member belongs. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
mdx (Calculated Member MDX Formula)	Specifies the MDX formula for the calculated member. [*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn2.microsoft.com/en-us/library/ms145595.aspx *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
memberName (OLAP Calculated Member Name)	Specifies the OLAP member name for the calculated member. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
name (Calculated Member Name)	Specifies the name of the calculated member. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
parent (Parent Name)	Specifies the name of the parent of the calculated member. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
set (Set)	Specifies a boolean value that indicates whether this calculated member describes a calculated set rather than a calculated member. A value of `1` or `true` indicates this is a calculated set. A value of `0` or `false` indicates this is a calculated member. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
solveOrder (Calculated Members Solve Order)	Specifies the order in which this calculated member is calculated in relation to other calculated members. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalculatedMember) is located in §A.2. *end note*]

18.10.1.11 calculatedMembers (Calculated Members)

Represents the collection of calculated members in an OLAP PivotTable.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<calculatedMembers count="1">
  <calculatedMember name="[Product].[Product Categories].[All
    Products].[Calculated Member]" mdx="'[Product].[Product Categories].[All
    Products].[Accessories]'" memberName="Calculated Member"
    hierarchy="[Product].[Product Categories]" parent="[Product].[Product
    Categories].[All Products]"/>
</calculatedMembers>
```

end example]

Attributes	Description
count (Calculated Members Count)	Specifies the number of calculated members. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CalculatedMembers) is located in §A.2. *end note*]

18.10.1.12 chartFormat (PivotChart Format)

Represents the format defined in the PivotChart that is associated with this PivotTable.

[*Example*:

```
<sh:pivotTableDefinition xmlns:sh="…" name="PivotTable1" cacheId="0"
  applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="0"
  applyPatternFormats="0" applyAlignmentFormats="0" applyWidthHeightFormats="1"
  dataCaption="Values" updatedVersion="3" minRefreshableVersion="3"
  showCalcMbrs="0" useAutoFormatting="1" colGrandTotals="0" itemPrintTitles="1"
  createdVersion="3" indent="0" outline="1" outlineData="1"
  multipleFieldFilters="0" chartFormat="1" fieldListSortAscending="1">
```

end example]

Attributes	Description
chart (Chart Index)	Specifies the index of the chart part to which the formatting applies. For more information see the DrawingML specification for more information on the chart part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
format (Pivot Format Id)	Specifies the index of the pivot format that is currently in use. This index corresponds to a dxf element in the Styles part. For more information see the Styles section (§18.8). The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
series (Series Format)	Specifies a boolean value that indicates whether format applies to a series. A value of `1` or `true` indicates this format applies to a series. A value of `0` or `false` indicates this format applies to a data point. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartFormat) is located in §A.2. *end note*]

18.10.1.13 chartFormats (PivotChart Formats)

Represents the collection of formats applied to PivotChart.

[*Example*:

```
<sh:chartFormats count="4">
  <sh:chartFormat chart="0" format="0" series="1">
    <sh:pivotArea type="data" outline="0">
      <sh:references count="3">
        <sh:reference field="4294967294" count="1" selected="0">
          <sh:x v="0"/>
        </sh:reference>
        <sh:reference field="14" count="1" selected="0">
          <sh:x v="0"/>
        </sh:reference>
        <sh:reference field="15" count="1" selected="0">
          <sh:x v="2"/>
        </sh:reference>
      </sh:references>
    </sh:pivotArea>
  </sh:chartFormat>
  <sh:chartFormat chart="0" format="1" series="1">
    <sh:pivotArea type="data" outline="0">
      <sh:references count="3">
        <sh:reference field="4294967294" count="1" selected="0">
          <sh:x v="0"/>
        </sh:reference>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
          <sh:reference field="14" count="1" selected="0">
            <sh:x v="0"/>
          </sh:reference>
          <sh:reference field="15" count="1" selected="0">
            <sh:x v="3"/>
          </sh:reference>
        </sh:references>
      </sh:pivotArea>
    </sh:chartFormat>
    <sh:chartFormat chart="0" format="2" series="1">
      <sh:pivotArea type="data" outline="0">
        <sh:references count="3">
          <sh:reference field="4294967294" count="1" selected="0">
            <sh:x v="1"/>
          </sh:reference>
          <sh:reference field="14" count="1" selected="0">
            <sh:x v="0"/>
          </sh:reference>
          <sh:reference field="15" count="1" selected="0">
            <sh:x v="2"/>
          </sh:reference>
        </sh:references>
      </sh:pivotArea>
    </sh:chartFormat>
    <sh:chartFormat chart="0" format="3" series="1">
      <sh:pivotArea type="data" outline="0">
        <sh:references count="3">
          <sh:reference field="4294967294" count="1" selected="0">
            <sh:x v="1"/>
          </sh:reference>
          <sh:reference field="14" count="1" selected="0">
            <sh:x v="0"/>
          </sh:reference>
          <sh:reference field="15" count="1" selected="0">
            <sh:x v="3"/>
          </sh:reference>
        </sh:references>
      </sh:pivotArea>
    </sh:chartFormat>
  </sh:chartFormats>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

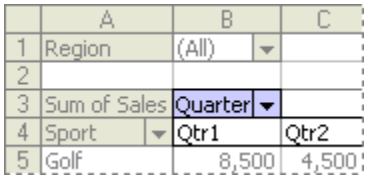
©ISO/IEC 2016 – All rights reserved

Attributes	Description
count (Format Count)	Specifies the number of formats in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartFormats) is located in §A.2. *end note*]

18.10.1.14 colFields (Column Fields)

Represents the collection of fields that are on the column axis of the PivotTable.



	A	B	C
1	Region	(All)	
2			
3	Sum of Sales	Quarter	
4	Sport	Qtr1	Qtr2
5	Golf	8,500	4,500

In the image above, the blue field is a column field.

[*Example*: In the following SpreadsheetML example, "Year", "Quarter" and "Month" are on the column axis of the PivotTable, in that order.

```
<colFields count="3">
  <field x="14"/>
  <field x="15"/>
  <field x="16"/>
</colFields>
```

end example]

Attributes	Description
count (Repeated Items Count)	Specifies the number of items in this collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColFields) is located in §A.2. *end note*]

18.10.1.15 colHierarchiesUsage (Column OLAP Hierarchy References)

Represents the collection of references to OLAP hierarchies on the column axis of a PivotTable.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<sh:colHierarchiesUsage count="2">
  <sh:colHierarchyUsage hierarchyUsage="33"/>
  <sh:colHierarchyUsage hierarchyUsage="-2"/>
</sh:colHierarchiesUsage>
```

end example]

Attributes	Description
count (Items Count)	Specifies the number of items in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColHierarchiesUsage) is located in §A.2. *end note*]

18.10.1.16 colHierarchyUsage (Column OLAP Hierarchies)

Represents the collection of references to OLAP Hierarchies on the column axis of a PivotTable.

[*Example*:

<sh:colHierarchyUsage hierarchyUsage="33"/>

end example]

Attributes	Description
hierarchyUsage (Hierarchy Usage)	Specifies the reference to an OLAP hierarchy in a PivotTable. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HierarchyUsage) is located in §A.2. *end note*]

18.10.1.17 colItems (Column Items)

Represents the collection of column items of the PivotTable.

[*Example*: In the following SpreadsheetML example the item values are found in cells C6:H8. For example "2001" / "3" / "July" values are in C7:C9. Those are the first column item values and are referenced by the first `<i>` element below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<colItems count="5">
  <i>
    <x/>
    <x/>
    <x/>
  </i>
  <i r="2">
    <x v="1"/>
  </i>
  <i r="2">
    <x v="2"/>
  </i>
  <i t="default" r="1">
    <x/>
  </i>
  <i t="default">
    <x/>
  </i>
</colItems>
```

end example]

The first <i> collection represents all item values for the first column in the column axis area of the PivotTable. The first <x> in the first <i> corresponds to the first field in the columns area of the PivotTable, namely "Year". The implied index value of '0' on this <x> indicates that the item value for this first item in the column is the 0th item for this pivotField. The 0th item for this pivotField is itself an index to an item value into this field's shared items collection in the pivotCacheDefinition part, namely "2001".

The item values corresponding to the second and third <x> elements can be found in the same way, arriving at "3" for the second item value, and arriving at "July" for the third item value for this first column.

The second <i> collection expresses all 3 item values for the second column in the column axis area. The @r value of '2' indicates that the first two item values from the previous column is repeated here, which means that the first item value for this second column is "2001" again and the second item value for this second column is "3". The third item value is expressed by the only <x> element under this second <i> element, and without further explanation is understood to reference the item value "August".

Attributes	Description
count (Column Item Count)	Specifies the number of items on the column axis of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_colItems) is located in §A.2. *end note*]

18.10.1.18 conditionalFormat (Conditional Formatting)

Represents the conditional formatting defined in the PivotTable.

Attributes	Description
priority (Priority)	Specifies the priority of PivotTable conditional formatting rule. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
scope (Conditional Formatting Scope)	Specifies the scope of PivotTable conditional formatting rule. The possible values for this attribute are defined by the ST_Scope simple type (§18.18.67).
type (Conditional Formatting Rule Type)	Specifies the type of PivotTable conditional formatting rule. See associated simple type definition for details. The possible values for this attribute are defined by the ST_Type simple type (§18.18.84).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConditionalFormat) is located in §A.2. *end note*]

18.10.1.19 conditionalFormats (Conditional Formats)

Represents the collection of conditional formats applied to a PivotTable.

[*Example*:

```
<sh:conditionalFormats count="1">
  <sh:conditionalFormat priority="1">
    <sh:pivotAreas count="1">
      <sh:pivotArea type="data" collapsedLevelsAreSubtotals="1">
        <sh:references count="5">
          <sh:reference field="4294967294" count="1" selected="0">
            <sh:x v="0"/>
          </sh:reference>
          <sh:reference field="2" count="1" selected="0">
            <sh:x v="0"/>
          </sh:reference>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <sh:reference field="14" count="1" selected="0">
              <sh:x v="0"/>
            </sh:reference>
            <sh:reference field="15" count="2" selected="0">
              <sh:x v="2"/>
              <sh:x v="3"/>
            </sh:reference>
          </sh:references>
        </sh:pivotArea>
      </sh:pivotAreas>
    </sh:conditionalFormat>
  </sh:conditionalFormats>
```

end example]

Attributes	Description
count (Conditional Format Count)	Specifies the number of conditional formats defined for the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConditionalFormats) is located in §A.2. *end note*]

18.10.1.20 consolidation (Consolidation Source)

Represents the description of the PivotCache source using multiple consolidation ranges. This element is used when the source of the PivotTable is a collection of ranges in the workbook. The ranges are specified in the rangeSets collection. The logic for how the application consolidates the data in the ranges is application-defined. [*Example*: the application might consolidate data based on its position in the worksheet that the end-user specifies. *end example*]

Attributes	Description
autoPage (Auto Page)	Specifies a boolean value that indicates whether the application will automatically create one additional page field to describe/qualify the source ranges. A value of `1` or `true` indicates the application will create an additional page field. A value of `0` or `false` indicates will not create an additional page field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Consolidation) is located in §A.2. *end note*]

18.10.1.21 d (Date Time)

Represents a date-time value in the PivotTable.

Attributes	Description
c (Caption)	Specifies the caption for the item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cp (Member Property Count)	Specifies the number of member property values. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item Value)	Specifies a boolean value that indicates whether this is a calculated item value. A value of `1` or `true` indicates this is a calculated item value. A value of `0` or `false` indicates this is not a calculated item value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of `1` or `true` indicates this is an unused item. A value of `0` or `false` indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
v (Value)	Specifies the value of the item. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.

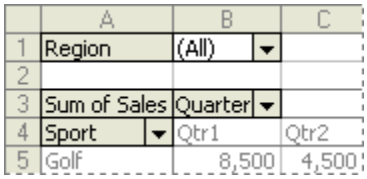
[*Note*: The W3C XML Schema definition of this element's content model (CT_DateTime) is located in §A.2. *end note*]

18.10.1.22 dataField (Data Field Item)

Represents a field from a source list, table, or database that contains data that is summarized in a PivotTable.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



	A	B	C
1	Region	(All)	
2			
3	Sum of Sales	Quarter	
4	Sport	Qtr1	Qtr2
5	Golf	8,500	4,500

A data field represents data that is derived from a field in the source list or database. [*Example*: The Sport field, for example, might come from a column in the source list that is labeled Sport and contains the names of various sports (Golf, Tennis) for which the source list has sales figures. *end example*]

Source data can be taken from an SpreadsheetML list or range, an external database or cube, or another PivotTable. Data fields use summary functions to combine values from the underlying source data. You can also use custom calculations to compare data values, or add your own formulas that use elements of the report or other worksheet data.

[*Example*:

```
<dataFields count="1">
  <dataField name="Sum of Sales Amount" fld="25" baseField="0" baseItem="0"/>
</dataFields>
```

end example]

Attributes	Description
baseField ('Show Data As' Base Field)	Specifies the index to the base field when the ShowDataAs calculation is in use. The possible values for this attribute are defined by the W3C XML Schema int datatype.
baseItem ('Show Data As' Base Setting)	Specifies the index to the base item when the ShowDataAs calculation is in use. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
fld (Field)	Specifies the index to the field (<r>) in the pivotCacheRecords part that this data item summarizes. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Data Field Name)	Specifies the name of the data field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
numFmtId (Number Format Id)	Specifies the index to the number format applied to this data field. Number formats are written to the styles part. See the Styles section(§18.8) for more information on number formats. Formatting information provided by cell table and by PivotTable need not agree. If the two formats differ, the cell-level formatting takes precedence. If you change the layout

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	the PivotTable, the PivotTable formatting will then take precedence. The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
showDataAs (Show Data As Display Format)	Specifies the display format for this data field. Formatting information provided by cell table and by PivotTable need not agree. If the two formats differ, the cell-level formatting takes precedence. If you change the layout the PivotTable, the PivotTable formatting will then take precedence. The possible values for this attribute are defined by the ST_ShowDataAs simple type (§18.18.70).
subtotal (Subtotal)	Specifies the aggregation function that applies to this data field. The possible values for this attribute are defined by the ST_DataConsolidateFunction simple type (§18.18.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataField) is located in §A.2. *end note*]

18.10.1.23 dataFields (Data Fields)

Represents the collection of items in the data region of the PivotTable.

[*Example*:

```
<dataFields count="1">
  <dataField name="Sum of Sales Amount" fld="25" baseField="0" baseItem="0"/>
</dataFields>
```

end example]

Attributes	Description
count (Data Items Count)	Specifies the number of items in the data region of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataFields) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.10.1.24 dimension (OLAP Dimension)

Represents a PivotTable OLAP Dimension. A dimension is a field that organizes a single type of data into a hierarchy with levels of detail. [*Example*: An OLAP database could contain a Time dimension providing data for levels Year, Month, Week, and Day, allowing you to create reports that let you compare day-to-day sales results or view a summary of your sales for an entire year. *end example*]

Attributes	Description
caption (Dimension Display Name)	Specifies the display name of the dimension. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
measure (Measure)	Specifies a boolean value that indicates whether this is a measure dimension. A value of `1` or `true` indicates this dimension is a measure dimension. A value of `0` or `false` indicates this dimension is not a measure dimension. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Dimension Name)	Specifies the name of the dimension. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
uniqueName (Dimension Unique Name)	Specifies the unique name of the dimension. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotDimension) is located in §A.2. *end note*]

18.10.1.25 dimensions (OLAP Dimensions)

Represents the collection of PivotTable OLAP dimensions.

[*Example*:

```
<dimensions count="22">
  <dimension name="Account" uniqueName="[Account]" caption="Account"/>
  <dimension name="Customer" uniqueName="[Customer]" caption="Customer"/>
  <dimension name="Date" uniqueName="[Date]" caption="Date"/>
  <dimension name="Delivery Date" uniqueName="[Delivery Date]"
    caption="Delivery Date"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <dimension name="Department" uniqueName="[Department]" caption="Department"/>
    <dimension name="Destination Currency" uniqueName="[Destination Currency]"
      caption="Destination Currency"/>
    <dimension name="Employee" uniqueName="[Employee]" caption="Employee"/>
    <dimension name="Geography" uniqueName="[Geography]" caption="Geography"/>
    <dimension name="Internet Sales Order Details" uniqueName="[Internet Sales
      Order Details]" caption="Internet Sales Order Details"/>
    <dimension measure="1" name="Measures" uniqueName="[Measures]"
      caption="Measures"/>
    <dimension name="Organization" uniqueName="[Organization]"
      caption="Organization"/>
    <dimension name="Product" uniqueName="[Product]" caption="Product"/>
    <dimension name="Promotion" uniqueName="[Promotion]" caption="Promotion"/>
    <dimension name="Reseller" uniqueName="[Reseller]" caption="Reseller"/>
    <dimension name="Reseller Sales Order Details" uniqueName="[Reseller Sales
      Order Details]" caption="Reseller Sales Order Details"/>
    <dimension name="Sales Channel" uniqueName="[Sales Channel]" caption="Sales
      Channel"/>
    <dimension name="Sales Reason" uniqueName="[Sales Reason]" caption="Sales
      Reason"/>
    <dimension name="Sales Summary Order Details" uniqueName="[Sales Summary Order
      Details]" caption="Sales Summary Order Details"/>
    <dimension name="Sales Territory" uniqueName="[Sales Territory]"
      caption="Sales Territory"/>
    <dimension name="Scenario" uniqueName="[Scenario]" caption="Scenario"/>
    <dimension name="Ship Date" uniqueName="[Ship Date]" caption="Ship Date"/>
    <dimension name="Source Currency" uniqueName="[Source Currency]"
      caption="Source Currency"/>
  </dimensions>
```

end example]

Attributes	Description
count (OLAP Dimensions Count)	Specifies the number of OLAP dimensions in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dimensions) is located in §A.2. *end note*]

18.10.1.26 discretePr (Discrete Grouping Properties)

Represents the collection of discrete grouping properties for a field group.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

```
…
  <fieldGroup par="6" base="0">
    <rangePr groupBy="months" startDate="2002-01-01T00:00:00"
      endDate="2006-05-06T00:00:00"/>
    <groupItems count="14">
      <s v="&lt;1/1/2002"/>
      <s v="Jan"/>
      <s v="Feb"/>
      <s v="Mar"/>
      <s v="Apr"/>
      <s v="May"/>
      <s v="Jun"/>
      <s v="Jul"/>
      <s v="Aug"/>
      <s v="Sep"/>
      <s v="Oct"/>
      <s v="Nov"/>
      <s v="Dec"/>
      <s v="&gt;5/6/2006"/>
    </groupItems>
  </fieldGroup>
</cacheField>
<cacheField name="Name" numFmtId="0">
  <sharedItems count="4">
    <s v="Joe"/>
    <s v="John"/>
    <s v="Bob"/>
    <s v="Robert"/>
  </sharedItems>
  <fieldGroup par="4"/>
</cacheField>
<cacheField name="ProductID" numFmtId="0">
  <sharedItems containsSemiMixedTypes="0" containsString="0" containsNumber="1"
    containsInteger="1" minValue="1" maxValue="4" count="4">
    <n v="1"/>
    <n v="2"/>
    <n v="3"/>
    <n v="4"/>
  </sharedItems>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <fieldGroup base="2">
      <rangePr startNum="1" endNum="4" groupInterval="2"/>
      <groupItems count="4">
        <s v="&lt;1"/>
        <s v="1-2"/>
        <s v="3-4"/>
        <s v="&gt;5"/>
      </groupItems>
    </fieldGroup>
  </cacheField>
```

end example]

Attributes	Description
count (Mapping Index Count)	Specifies the number of mapping indexes for this grouped field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DiscretePr) is located in §A.2. *end note*]

18.10.1.27 e (Error Value)

Represents an error value. The use of this item indicates that an error value is present in the PivotTable source. The error is recorded in the `value` attribute.

Attributes	Description
b (Bold)	Specifies a boolean value that indicates whether the value contains bold formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of 1 or `true` indicates this value contains bold formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bc (background Color)	Specifies the background color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
c (Item Caption)	Specifies the item/member caption The possible values for this attribute are defined by the ST_Xstring simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
cp (Member Property Count)	Specifies the number of member property values. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item)	Specifies a boolean value that indicates whether this is a calculated item value. A value of `1` or `true` indicates value is a calculated item value. A value of `0` or `false` indicates this value is not a calculated item value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fc (Foreground Color)	Specifies the foreground color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
i (Italic)	Specifies a boolean value that indicates whether the value contains italic formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains italic formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
in (Format Index)	Specifies the index to the OLAP serverformat element where the format string for this entry is stored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
st (Strikethrough)	Specifies a boolean value that indicates whether the value contains strikethrough formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains strikethrough formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of `1` or `true` indicates this item is not used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	A value of `0` or `false` indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
un (Underline)	Specifies a boolean value that indicates whether the value contains underline formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
v (Value)	Specifies the value of the item. This attribute depends on how the application records errors. [*Note*: While the error values are determined by the application, the following are some example error values that could be used: • #DIV/0! • #NAME? • #VALUE! • #NULL! • #NUM! • #REF! • #N/A • #GETTING_DATA *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Error) is located in §A.2. *end note*]

18.10.1.28 entries (Entries)

Represents the collection of OLAP sheet data entries.

Attributes	Description
count (Tuple Count)	Specifies the number of tuple entries. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PCDSDTCEntries) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.29 field (Field)

Represents a generic field that can appear either on the column or the row region of the PivotTable. There areas many <x> elements as there are item values in any particular column or row.

[*Example*:

```
<sh:field x="2"/>
```

end example]

Attributes	Description
x (Field Index)	Specifies the index to a pivotField item value. There are as many x elements as there are item values in any particular column. Note that these x elements sometimes are not explicitly written, but instead "inherited" from the previous column or i element, via the value of @r. The pivotField items don't list values explicitly, but instead reference a shared item value in the pivotCacheDefinition part. The first instance of x has no attribute value @v associated with it, so the default value for @v is assumed to be "0". The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Field) is located in §A.2. *end note*]

18.10.1.30 fieldGroup (Field Group Properties)

Represents the collection of properties for a field group.

[*Example*:

```
…
  <fieldGroup par="6" base="0">
    <rangePr groupBy="months" startDate="2002-01-01T00:00:00"
      endDate="2006-05-06T00:00:00"/>
    <groupItems count="14">
      <s v="&lt;1/1/2002"/>
      <s v="Jan"/>
      <s v="Feb"/>
      <s v="Mar"/>
      <s v="Apr"/>
      <s v="May"/>
      <s v="Jun"/>
      <s v="Jul"/>
      <s v="Aug"/>
      <s v="Sep"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <s v="Oct"/>
        <s v="Nov"/>
        <s v="Dec"/>
        <s v="&gt;5/6/2006"/>
      </groupItems>
    </fieldGroup>
  </cacheField>
  <cacheField name="Name" numFmtId="0">
    <sharedItems count="4">
      <s v="Joe"/>
      <s v="John"/>
      <s v="Bob"/>
      <s v="Robert"/>
    </sharedItems>
    <fieldGroup par="4"/>
  </cacheField>
  <cacheField name="ProductID" numFmtId="0">
    <sharedItems containsSemiMixedTypes="0" containsString="0" containsNumber="1"
      containsInteger="1" minValue="1" maxValue="4" count="4">
      <n v="1"/>
      <n v="2"/>
      <n v="3"/>
      <n v="4"/>
    </sharedItems>
    <fieldGroup base="2">
      <rangePr startNum="1" endNum="4" groupInterval="2"/>
      <groupItems count="4">
        <s v="&lt;1"/>
        <s v="1-2"/>
        <s v="3-4"/>
        <s v="&gt;5"/>
      </groupItems>
    </fieldGroup>
  …
```

end example]

Attributes	Description
base (Field Base)	Specifies the base of this field, if any. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
par (Parent)	Specifies the parent of this field, if any.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FieldGroup) is located in §A.2. *end note*]

18.10.1.31 fieldsUsage (Fields Usage)

Represents the fields in the cache that are being used by this hierarchy.

[*Example*:

```
<fieldsUsage count="6">
  <fieldUsage x="-1"/>
  <fieldUsage x="2"/>
  <fieldUsage x="3"/>
  <fieldUsage x="4"/>
  <fieldUsage x="5"/>
  <fieldUsage x="6"/>
</fieldsUsage>
```

end example]

Attributes	Description
count (Field Count)	Specifies the number of fields that are being used by this hierarchy. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FieldsUsage) is located in §A.2. *end note*]

18.10.1.32 fieldUsage (PivotCache Field Id)

Represents a cache field used in this hierarchy.

[*Example*:

```
<fieldUsage x="-1"/>
```

end example]

Attributes	Description
x (Field Index)	Specifies the index of a field.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FieldUsage) is located in §A.2. *end note*]

18.10.1.33 filter (PivotTable Advanced Filter)

Represents a PivotTable advanced filter.

[*Example*:

```
<sh:filter fld="3" type="count" id="1" iMeasureHier="187">
  <sh:autoFilter ref="A1">
    <sh:filterColumn colId="0">
      <sh:top10 val="5"/>
    </sh:filterColumn>
  </sh:autoFilter>
</sh:filter>
```

end example]

Attributes	Description
description (Pivot Filter Description)	Specifies the description of the pivot filter. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
evalOrder (Evaluation Order)	Specifies the evaluation order of the pivot filter. This attribute is zero-based. The possible values for this attribute are defined by the W3C XML Schema int datatype.
fld (Field Index)	Specifies the index of the field to which this pivot filter belongs. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Pivot Filter Id)	Specifies the unique identifier of the pivot filter as assigned by the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
iMeasureFld (Measure Field Index)	Specifies the index of the measure field. This attribute is used only by filters in Relational pivots and specifies on which measure a value filter should apply. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
iMeasureHier (Measure Index)	Specifies the index of the measure cube field. This attribute is used only by filters in OLAP pivots and specifies on which measure a value filter should apply. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
mpFld (Member Property Field Id)	Specifies the index of the field representing the member property field on which this pivot filter is defined. This attribute is used only by label pivot filters. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Pivot Filter Name)	Specifies the name of the pivot filter. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
stringValue1 (Label Pivot)	Specifies the string value "1" used by label pivot filters. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
stringValue2 (Label Pivot Filter String Value 2)	Specifies the string value "2" used by label pivot filters. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
type (Pivot Filter Type)	Specifies the type of the pivot filter. The possible values for this attribute are defined by the ST_PivotFilterType simple type (§18.18.59).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFilter) is located in §A.2. *end note*]

18.10.1.34 filters (Filters)

Represents the collection of filters that apply to this PivotTable.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<sh:filters count="1">
  <sh:filter fld="3" type="count" id="1" iMeasureHier="187">
    <sh:autoFilter ref="A1">
      <sh:filterColumn colId="0">
        <sh:top10 val="5"/>
      </sh:filterColumn>
    </sh:autoFilter>
  </sh:filter>
</sh:filters>
```

end example]

Attributes	Description
count (Pivot Filter Count)	Specifies the number of pivot filters in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFilters) is located in §A.2. *end note*]

18.10.1.35 format (PivotTable Format)

Represents the format defined in the PivotTable.

Attributes	Description
action (Format Action)	Specifies the formatting behavior for the area indicated in the pivotArea element. The default value for this attribute is "formatting," which indicates that the specified cells have some formatting applied. The format is specified in the dxfId attribute. If the formatting is cleared from the cells, then the value of this attribute becomes "blank." The possible values for this attribute are defined by the ST_FormatAction simple type (§18.18.34).
dxfId (Format Id)	Specifies the identifier of the format the application is currently using for the PivotTable. Formatting information is written to the styles part. See the Styles section (§18.8) for more information on formats. Formatting information provided by cell table and by PivotTable need not agree. If the two formats differ, the cell-level formatting takes precedence. If you change the layout the PivotTable, the PivotTable formatting will then take precedence. The possible values for this attribute are defined by the ST_DxfId simple type (§18.18.25).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Format) is located in §A.2. *end note*]

18.10.1.36 formats (PivotTable Formats)

Represents the collection of formats applied to PivotTable.

Attributes	Description
count (Formats Count)	Specifies the number of formats in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Formats) is located in §A.2. *end note*]

18.10.1.37 group (OLAP Group)

Represents an OLAP level group.

[*Example*:

```
<group name="CategoryXl_Grp_1" uniqueName="[Product].[Product Categories].
  [Product Categories1].[GROUPMEMBER.[CategoryXl_Grp_1]].[Product]].
  [Product Categories]].[All Products]]" caption="Group1"
  uniqueParent="[Product].[Product Categories].[All Products]" id="1">
  <groupMembers count="2">
    <groupMember
      uniqueName="[Product].[Product Categories].[Category].&amp;[4]"/>
    <groupMember
      uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
  </groupMembers>
</group>
```

end example]

Attributes	Description
caption (Group Caption)	Specifies the caption for this group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
id (Group Id)	Specifies the unique number for this group within the level. The possible values for this attribute are defined by the W3C XML Schema int datatype.
name (Group	Specifies the name of this group.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Name)	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
uniqueName (Unique Group Name)	Specifies the unique name of this group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
uniqueParent (Parent Unique Name)	Specifies the unique name of the parent of this group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LevelGroup) is located in §A.2. *end note*]

18.10.1.38 groupItems (OLAP Group Items)

Represents the collection of items in a field group.

[*Example*:

```
…
  <fieldGroup par="6" base="0">
    <rangePr groupBy="months" startDate="2002-01-01T00:00:00"
      endDate="2006-05-06T00:00:00"/>
    <groupItems count="14">
      <s v="&lt;1/1/2002"/>
      <s v="Jan"/>
      <s v="Feb"/>
      <s v="Mar"/>
      <s v="Apr"/>
      <s v="May"/>
      <s v="Jun"/>
      <s v="Jul"/>
      <s v="Aug"/>
      <s v="Sep"/>
      <s v="Oct"/>
      <s v="Nov"/>
      <s v="Dec"/>
      <s v="&gt;5/6/2006"/>
    </groupItems>
  </fieldGroup>
</cacheField>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<cacheField name="Name" numFmtId="0">
  <sharedItems count="4">
    <s v="Joe"/>
    <s v="John"/>
    <s v="Bob"/>
    <s v="Robert"/>
  </sharedItems>
  <fieldGroup par="4"/>
</cacheField>
<cacheField name="ProductID" numFmtId="0">
  <sharedItems containsSemiMixedTypes="0" containsString="0" containsNumber="1"
    containsInteger="1" minValue="1" maxValue="4" count="4">
    <n v="1"/>
    <n v="2"/>
    <n v="3"/>
    <n v="4"/>
  </sharedItems>
  <fieldGroup base="2">
    <rangePr startNum="1" endNum="4" groupInterval="2"/>
    <groupItems count="4">
      <s v="&lt;1"/>
      <s v="1-2"/>
      <s v="3-4"/>
      <s v="&gt;5"/>
    </groupItems>
  </fieldGroup>
…
```

end example]

Attributes	Description
count (Items Created Count)	Specifies the number of items created for this grouped field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupItems) is located in §A.2. *end note*]

18.10.1.39 groupLevel (OLAP Grouping Levels)

Represents the collection of OLAP grouping levels.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<groupLevel uniqueName="[Product].[Product Categories].[Category]"
  caption="Category">
  <groups count="1">
    <group name="CategoryXl_Grp_1" uniqueName="[Product].[Product
      Categories].[Product Categories1].
      [GROUPMEMBER.[CategoryXl_Grp_1]].[Product]].[Product Categories]].
      [All Products]]]" caption="Group1" uniqueParent="[Product].
      [Product Categories].[All Products]" id="1">
      <groupMembers count="2">
        <groupMember
          uniqueName="[Product].[Product Categories].[Category].&amp;[4]"/>
        <groupMember
          uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
      </groupMembers>
    </group>
  </groups>
</groupLevel>
```

end example]

Attributes	Description
caption (Grouping Level Display Name)	Specifies the display name for this grouping level. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
customRollUp (Custom Roll Up)	Specifies a boolean value that indicates whether this group level has a custom roll up. A value of `1` or `true` indicates this group level has a custom roll up. A value of `0` or `false` indicates this group level does not have a custom roll up. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uniqueName (Unique Name)	Specifies the unique name for this grouping level. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
user (User-Defined Group Level)	Specifies a boolean value that indicates whether this is a user-defined group level. A value of `1` or `true` indicates this is a user-defined group. A value of `0` or `false` indicates this group is not user-defined. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupLevel) is located in §A.2. *end note*]

18.10.1.40 groupLevels (OLAP Grouping Levels)

Represents the collection of OLAP grouping levels.

[*Example*:

```
<groupLevels count="5">
  <groupLevel uniqueName="[Product].[Product Categories].[(All)]"
    caption="(All)"/>
  <groupLevel uniqueName="[Product].[Product Categories].[Product Categories1]"
    caption="Product Categories1" user="1"/>
  <groupLevel uniqueName="[Product].[Product Categories].[Category]"
    caption="Category">
    <groups count="1">
      <group name="CategoryXl_Grp_1" uniqueName="[Product].[Product Categories].
        [Product Categories1].[GROUPMEMBER.[CategoryXl_Grp_1]].
        [Product]].[Product Categories]].[All Products]]]" caption="Group1"
        uniqueParent="[Product].[Product Categories].[All Products]" id="1">
        <groupMembers count="2">
          <groupMember
            uniqueName="[Product].[Product Categories].[Category].&amp;[4]"/>
          <groupMember
            uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
        </groupMembers>
      </group>
    </groups>
  </groupLevel>
  <groupLevel uniqueName="[Product].[Product Categories].[Subcategory]"
    caption="Subcategory"/>
  <groupLevel uniqueName="[Product].[Product Categories].[Product]"
    caption="Product"/>
</groupLevels>
```

end example]

Attributes	Description
count (Grouping Level Count)	Specifies the number of grouping levels. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupLevels) is located in §A.2. *end note*]

18.10.1.41 groupMember (OLAP Group Member)

Represents an OLAP group member.

[*Example*:

```
<groupMember uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
```

end example]

Attributes	Description
group (Group)	Specifies a boolean value that indicates whether this member represents a group. A value of `1` or `true` indicates this member represents a group. A value of `0` or `false` indicates this member does not represent a group. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uniqueName (Group Member Unique Name)	Specifies the unique name of this group member. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupMember) is located in §A.2. *end note*]

18.10.1.42 groupMembers (OLAP Group Members)

Represents the collection of OLAP group members.

[*Example*:

```
<groupMembers count="2">
  <groupMember uniqueName="[Product].[Product Categories].[Category].&amp;[4]"/>
  <groupMember uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
</groupMembers>
```

end example]

Attributes	Description
count (Group Member Count)	Specifies the number of group members in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupMembers) is located in §A.2. *end note*]

18.10.1.43 groups (OLAP Level Groups)

Represents the collection of OLAP level groups.

[*Example*:

```
<groups count="1">
  <group name="CategoryXl_Grp_1" uniqueName="[Product].[Product Categories].
    [Product Categories1].[GROUPMEMBER.[CategoryXl_Grp_1]].[Product]].
    [Product Categories]].[All Products]]]" caption="Group1"
    uniqueParent="[Product].[Product Categories].[All Products]" id="1">
    <groupMembers count="2">
      <groupMember
        uniqueName="[Product].[Product Categories].[Category].&amp;[4]"/>
      <groupMember
        uniqueName="[Product].[Product Categories].[Category].&amp;[1]"/>
    </groupMembers>
  </group>
</groups>
```

end example]

Attributes	Description
count (Level Group Count)	Specifies the number of level groups in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Groups) is located in §A.2. *end note*]

18.10.1.44 i (Row Items)

Represents the collection of items in the row region of the PivotTable.

[*Example*: In this example the item values are found in cells B10:B13. For example "Bikes" is in B10, and corresponds to the first <i> element below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<rowItems count="4">
  <i>
    <x/>
  </i>
  <i r="1">
    <x/>
  </i>
  <i r="1">
    <x v="1"/>
  </i>
  <i t="grand">
    <x/>
  </i>
</rowItems>
```

end example]

Attributes	Description
i (Data Field Index)	Specifies a zero-based index indicating the referenced data item it in a data field with multiple data items. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
r (Repeated Items Count)	Specifies the number of items to repeat from the previous row item. The first item has no @r explicitly written. Since a default of "0" is specified in the schema, for any item whose @r is missing, a default value of "0" is implied. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
t (Item Type)	Specifies the type of the item. Value of 'default' indicates a grand total as the last row item value The possible values for this attribute are defined by the ST_ItemType simple type (§18.18.43).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_I) is located in §A.2. *end note*]

18.10.1.45 item (PivotTable Field Item)

Represents a single item in PivotTable field.

[*Example*:

```
<sh:item x="66"/>
```


©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
c (Child Items)	Specifies a boolean value that indicates whether the approximate number of child items for this item is greater than zero. A value of `1` or `true` indicates the approximate number of child items for this item is greater than zero. A value of `0` or `false` indicates the approximate number of child items for this item is zero. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
d (Expanded)	Specifies a boolean value that indicates whether this item has been expanded in the PivotTable view. A value of `1` or `true` indicates this item has been expanded. A value of `0` or `false` indicates this item is collapsed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
e (Drill Across Attributes)	Specifies a boolean value that indicates whether attribute hierarchies nested next to each other on a PivotTable row or column will offer drilling "across" each other or not. [*Example*: if the application offers drill across for attribute hierarchies and not for user hierarchies, this attribute would only be written when two attribute hierarchies are placed next to each other on an axis. *end example*] A value of `1` or `true` indicates there is a drill across attribute hierarchies positioned next to each other on a pivot axis. A value of `0` or `false` indicates there is not drill across attribute hierarchies. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
f (Calculated Member)	Specifies a boolean value that indicates whether this item is a calculated member. A value of `1` or `true` indicates this item is a calculated member. A value of `0` or `false` indicates this item is not calculated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
h (Hidden)	Specifies a boolean value that indicates whether the item is hidden.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of 1 or true indicates item is hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
m (Missing)	Specifies a boolean value that indicate whether the item has a missing value. A value of 1 or true indicates the item value is missing. The application should still retain the item settings in case the item reappears during a later refresh. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
n (Item User Caption)	Specifies the user caption of the item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
s (Character)	Specifies a boolean value that indicates whether the item has a character value. A value of 1 or true indicates the item has a string/character value. A value of 0 or false indicates item the item has a value of a different type. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sd (Hide Details)	Specifies a boolean value that indicates whether the details are hidden for this item. A value of 1 or true indicates item details are hidden. A value of 0 or false indicates item details are shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
t (Item Type)	Specifies the type of this item. A value of 'default' indicates the subtotal or total item. The possible values for this attribute are defined by the ST_ItemType simple type (§18.18.43).

x (Item Index)	Specifies the item index in pivotFields collection in the PivotCache.
	
	[*Example*: In the following example, "Product Category" and "Product Subcategory" are on the row axis of the PivotTable, in that order.
	
	`<rowFields count="2">`
	`<field x="7"/>`
	`<x="8"/>`
	`</rowFieldsfield >`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

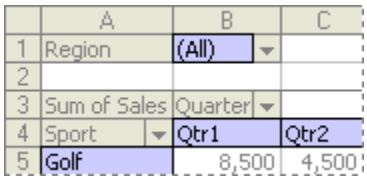
Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Item) is located in §A.2. *end note*]

18.10.1.46 items (Field Items)

Represents the collection of items in a PivotTable field. The items in the collection are ordered by index. Items represent the unique entries from the field in the source data.

In the following image, the item Golf represents all rows of data in the source list for which the Sport field contains the entry Golf.



The order in which the items are listed is the order they would appear on a particular axis [*Example*: Row or column. *end example*]

[*Example*: In the following SpreadsheetML example, the first field is "Customer Name" and the first item referenced here is <item x="66"/>, which references the value "Adam L Flores" in the pivotCacheDefinition. Therefore, if you added "Customer Name" to the row axis, "Adam L Flores" would be the first row item listed.

```
<pivotFields count="28">
  <pivotField showAll="0" includeNewItemsInFilter="1">
    <items count="8">
      <item x="66"/>
      <item x="133"/>
      <item x="74"/>
      <item x="27"/>
      <item x="118"/>
      <item x="63"/>
      <item x="141"/>
      <item t="default"/>
    </items>
  </pivotField>
  <pivotField showAll="0" includeNewItemsInFilter="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <pivotField axis="axisPage" showAll="0" includeNewItemsInFilter="1">
      <items count="2">
        <item x="0"/>
        <item t="default"/>
      </items>
    </pivotField>
  <pivotField showAll="0" includeNewItemsInFilter="1"/>
```

end example]

Attributes	Description
count (Field Count)	Specifies the number of fields in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Items) is located in §A.2. *end note*]

18.10.1.47 kpi (OLAP KPI)

Represents the KPI defined on the OLAP server and stored in the PivotCache.

[*Example*:

```
<kpi uniqueName="Growth in Customer Base" caption="Growth in Customer Base"
  displayFolder="Customer Perspective\Expand Customer Base"
  measureGroup="Internet Sales" value="[Measures].[Growth in Customer Base]"
  goal="[Measures].[Growth in Customer Base Goal]"
  status="[Measures].[Growth in Customer Base Status]"
  trend="[Measures].[Growth in Customer Base Trend]"/>
```

end example]

Attributes	Description
caption (KPI Display Name)	Specifies the display name of the KPI. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
displayFolder (KPI Display Folder)	Specifies the folder where this KPI is displayed in a list of fields for the PivotTable. This attribute depends on how the application exposes a list of fields in the user interface. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
goal (KPI Goal Unique Name)	Specifies the unique name of the KPI goal measure.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
measureGroup (KPI Measure Group Name)	Specifies the name of the measure group to which this KPI belongs. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
parent (Parent KPI)	Specifies the name of the parent KPI for this KPI. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
status (KPI Status Unique Name)	Specifies the unique name of the KPI status measure. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
time (Time Member KPI Unique Name)	Specifies the unique name of the KPI current time member. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
trend (KPI Trend Unique Name)	Specifies the unique name of the KPI trend measure. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
uniqueName (KPI Unique Name)	Specifies the unique name of the KPI. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
value (KPI Value Unique Name)	Specifies the unique name of the KPI value measure. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
weight (KPI Weight Unique Name)	Specifies the unique name of the KPI weight measure. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PCDKPI) is located in §A.2. *end note*]

18.10.1.48 kpis (OLAP KPIs)

Represents the collection of Key Performance Indicators (KPIs) defined on the OLAP server and stored in the PivotCache.

[*Example*:

```
<kpis count="3">
  <kpi uniqueName="Growth in Customer Base" caption="Growth in Customer Base"
    displayFolder="Customer Perspective\Expand Customer Base"
    measureGroup="Internet Sales" value="[Measures].[Growth in Customer Base]"
    goal="[Measures].[Growth in Customer Base Goal]"
    status="[Measures].[Growth in Customer Base Status]"
    trend="[Measures].[Growth in Customer Base Trend]"/>
  <kpi uniqueName="Net Income" caption="Net Income"
    displayFolder="Financial Perspective\Maintain Overall Margins"
    measureGroup="Financial Reporting" value="[Measures].[Net Income Value]"
    goal="[Measures].[Net Income Goal]" status="[Measures].[Net Income Status]"
    trend="[Measures].[Net Income Trend]"/>
  <kpi uniqueName="Operating Profit" caption="Operating Profit"
    displayFolder="Financial Perspective\Maintain Overall Margins"
    measureGroup="Financial Reporting" parent="Net Income"
    value="[Measures].[Operating Profit Value]"
    goal="[Measures].[Operating Profit Goal]"
    status="[Measures].[Operating Profit Status]"
    trend="[Measures].[Operating Profit Trend]"/>
…
```

end example]

Attributes	Description
count (KPI Count)	Specifies the number of KPIs stored in the PivotCache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PCDKPIs) is located in §A.2. *end note*]

18.10.1.49 location (PivotTable Location)

Represents location information for the PivotTable.

[*Example*:

```
<location ref="B6:G13" firstHeaderRow="1" firstDataRow="4" firstDataCol="1"
  rowPageCount="3" colPageCount="1"/>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
colPageCount (Columns Per Page)	Specifies the number of columns per page for this PivotTable that the filter area will occupy. By default there is a single column of filter fields per page and the fields occupy as many rows as there are fields. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
firstDataCol (First Data Column)	Specifies the first column of the PivotTable data, relative to the top left cell in the `ref` value. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
firstDataRow (PivotTable Data First Row)	Specifies the first row of the PivotTable data, relative to the top left cell in the `ref` value. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
firstHeaderRow (First Header Row)	Specifies the first row of the PivotTable header, relative to the top left cell in the `ref` value. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ref (Reference)	Specifies the first row of the PivotTable. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
rowPageCount (Rows Per Page Count)	Specifies the number of rows per page for this PivotTable that the filter area will occupy. By default there is a single column of filter fields per page and the fields occupy as many rows as there are fields. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Location) is located in §A.2. *end note*]

18.10.1.50 m (No Value)

Represents a value that was not specified.

[*Example*:

```
<sharedItems containsString="0" containsBlank="1" count="1">
  <m/>
</sharedItems>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
b (Bold)	Specifies a boolean value that indicates whether the value contains bold formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains bold formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bc (background Color)	Specifies the background color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
c (Caption)	Specifies the caption for this item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cp (Member Property Count)	Specifies the number of member property values for this item. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item)	Specifies a boolean value that indicates whether this is a calculated item value. A value of `1` or `true` indicates this item is a calculated value. A value of `0` or `false` indicates this item is not calculated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fc (Foreground Color)	Specifies the foreground color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
i (Italic)	Specifies a boolean value that indicates whether the value contains italic formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
in (Format Index)	Specifies the index to the OLAP serverformat element where the format string for this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	entry is stored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
st (Strikethrough)	Specifies a boolean value that indicates whether the value contains strikethrough formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of `1` or `true` indicates this item is unused. A value of `0` or `false` indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
un (Underline)	Specifies a boolean value that indicates whether the value contains underline formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Missing) is located in §A.2. *end note*]

18.10.1.51 map (OLAP Measure Group)

Represents a PivotTable OLAP measure group - Dimension map.

[*Example*:

```
<map measureGroup="0" dimension="2"/>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
dimension (Dimension Id)	Specifies the identifier for the dimension. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
measureGroup (Measure Group Id)	Specifies the identifier of the measure group. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MeasureDimensionMap) is located in §A.2. *end note*]

18.10.1.52 maps (OLAP Measure Group)

Represents the PivotTable OLAP measure group - Dimension maps.

[*Example*:

```
<maps count="3">
  <map measureGroup="0" dimension="2"/>
  <map measureGroup="1" dimension="19"/>
  <map measureGroup="2" dimension="8"/>
</maps>
```

end example]

Attributes	Description
count (Measure Group Count)	Specifies the number of measure groups, or dimension maps, in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MeasureDimensionMaps) is located in §A.2. *end note*]

18.10.1.53 measureGroup (OLAP Measure Group)

Represents a PivotTable OLAP measure group.

[*Example*:

```
<measureGroup name="Sales Orders" caption="Sales Orders"/>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
caption (Measure Group Display Name)	Specifies the display name of the measure group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
name (Measure Group Name)	Specifies the name of the measure group. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MeasureGroup) is located in §A.2. *end note*]

18.10.1.54 measureGroups (OLAP Measure Groups)

Represents the collection of PivotTable OLAP measure groups.

[*Example*:

```
<measureGroups count="11">
  <measureGroup name="Exchange Rates" caption="Exchange Rates"/>
  <measureGroup name="Financial Reporting" caption="Financial Reporting"/>
  <measureGroup name="Internet Customers" caption="Internet Customers"/>
  <measureGroup name="Internet Orders" caption="Internet Orders"/>
  <measureGroup name="Internet Sales" caption="Internet Sales"/>
  <measureGroup name="Reseller Orders" caption="Reseller Orders"/>
  <measureGroup name="Reseller Sales" caption="Reseller Sales"/>
  <measureGroup name="Sales Orders" caption="Sales Orders"/>
  <measureGroup name="Sales Reasons" caption="Sales Reasons"/>
  <measureGroup name="Sales Summary" caption="Sales Summary"/>
  <measureGroup name="Sales Targets" caption="Sales Targets"/>
</measureGroups>
```

end example]

Attributes	Description
count (Measure Group Count)	Specifies the number of measure groups in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MeasureGroups) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.10.1.55 member (Member)

Represents an item that can be included or excluded.

Attributes	Description
name (Hidden Item Name)	Specifies the name of a hidden item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Member) is located in §A.2. *end note*]

18.10.1.56 members (Members)

Represents the collection of items that can be included or excluded.

Attributes	Description
count (Item Count)	Specifies the number of items in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
level (Hierarchy Level)	Specifies the hierarchy level with which these items are associated. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Members) is located in §A.2. *end note*]

18.10.1.57 mp (OLAP Member Property)

Represents an OLAP member property.

[*Example*:

```
<sh:mp field="7"/>
```

end example]

Attributes	Description
field (Field Index)	Specifies the index of the field with which this member property is associated. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
level (Level Index)	Specifies the index of the level to which this member property applies. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (OLAP Member Property Unique Name)	Specifies the unique name of the OLAP member property. The following attributes depend on the name attribute: • nameLen • pLen • pPos These attributes consist of metadata about a member in an OLAP cube and are usually displayed in a tooltip or mechanism in the user interface. [*Example*: If the value for name equals "[Store].[Store Name].[Store Manager]": • nameLen will equal 20. This would refer to “[Store].[Store Name]” • pPos will equal 22. This would refer to starting character of “Store Manager” • pLen will equal 13. This would to length of “Store Manager” *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
nameLen (Name Length)	Specifies the length of the unique name portion of name. [*Example*: If the value for name equals "[Store].[Store Name].[Store Manager]", nameLen will equal 20. This would refer to “[Store].[Store Name]”. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
pLen (Property Name Length)	Specifies the length of the property name portion of name. [*Example*: If the value for name equals "[Store].[Store Name].[Store Manager]", pLen will equal 13. This would to length of “Store Manager”. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
pPos (Property Name Character Index)	Specifies the index of the character where the property name portion begins in name. [*Example*: If the value for name equals "[Store].[Store Name].[Store Manager]", pPos will equal 22. This would refer to starting character of “Store Manager”. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
showAsCaption (Show As Caption)	Specifies a boolean value that indicates whether to show the property a member caption. A value of 1 or `true` indicates member property value is shown in as a caption.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	A value of `0` or `false` indicates member property value will not be shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showCell (Show Cell)	Specifies a boolean value that indicates whether to show the member property value in a PivotTable cell. A value of `1` or `true` indicates member property value is shown in a cell. A value of `0` or `false` indicates member property value will not be shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showTip (Show Tooltip)	Specifies a boolean value that indicates whether to show the member property value in a tooltip on the appropriate PivotTable view cells. A value of `1` or `true` indicates member property value is shown in a tooltip. A value of `0` or `false` indicates member property value will not be shown. This attribute depends on whether the application employs tooltips or similar mechanism in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MemberProperty) is located in §A.2. *end note*]

18.10.1.58 mpMap (Member Properties Map)

Represents a mapping to cached member properties.

[*Example*:

```
<mpMap v="7"/>
```

end example]

Attributes	Description
v (Shared Items Index)	Specifies the index into the shared items table in the PivotCache that identifies this item. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_X) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.59 mps (OLAP Member Properties)

Represents the collection of OLAP member property. Member properties contain additional information that is available about the items in an OLAP dimension field. [*Example*: If a Geography dimension has property fields Population and Average Income available, you could create a PivotTable report that displays the sales figures for cities where your products are selling well. By displaying and analyzing the population and income figures for these cities, you could target cities with similar demographics for your marketing campaign. *end example*]

[*Example*:

```
<sh:mps count="3">
  <sh:mp field="7"/>
  <sh:mp field="8"/>
  <sh:mp field="9"/>
</sh:mps>
```

end example]

Attributes	Description
count (OLAP Member Properties Count)	Specifies the number of OLAP member properties in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MemberProperties) is located in §A.2. *end note*]

18.10.1.60 n (Numeric)

Represents a numeric value in the PivotTable.

[*Example*:

```
<sharedItems containsSemiMixedTypes="0" containsString="0" containsNumber="1"
  containsInteger="1" minValue="3" maxValue="3" count="1">
  <n v="3"/>
</sharedItems>
```

end example]

Attributes	Description
b (Bold)	Specifies a boolean value that indicates whether this value contains bold formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of 1 or `true` indicates this value contains italic formatting on the server.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bc (Background Color)	Specifies the background color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
c (Caption)	Specifies the caption for this item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cp (Member Property Count)	Specifies the number of member property values for this item. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item)	Specifies a boolean value that indicates whether this is a calculated item value. A value of `1` or `true` indicates this item is a calculated value. A value of `0` or `false` indicates this item is not calculated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fc (Foreground Color)	Specifies the foreground color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
i (Italic)	Specifies a boolean value that indicates whether the value contains italic formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains italic formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
in (Format Index)	Specifies the index to the OLAP serverformat element where the format string for this entry is stored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
st (Strikethrough)	Specifies a boolean value that indicates whether the value contains strikethrough

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains strikethrough formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of `1` or `true` indicates this item is not used. A value of `0` or `false` indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
un (Underline)	Specifies a boolean value that indicates whether the value contains underline formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
v (Value)	Specified the value of this item. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Number) is located in §A.2. *end note*]

18.10.1.61 page (Page Items)

Represents the collection of page item values for a page field.

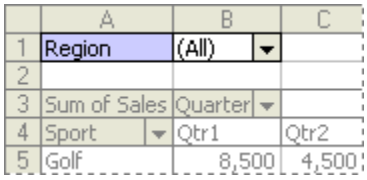
Attributes	Description
count (Page Item String Count)	Specifies the number of page item strings in the collectoin. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PCDSCPage) is located in §A.2. *end note*]

18.10.1.62 pageField (Page Field)

Represents a field on the page or report filter of the PivotTable.



	A	B	C
1	Region	(All)	
2			
3	Sum of Sales	Quarter	
4	Sport	Qtr1	Qtr2
5	Golf	8,500	4,500

In the image above, the blue field is a page or report filter field. Page/filter fields allow you to filter the entire PivotTable to display data for a single item or all the items.

[*Example*:

```
<sh:pageField fld="43" hier="103"
  name="[Product].[Product Categories].[All Products]" cap="All Products"/>
```

end example]

Attributes	Description
cap (Hierarchy Display Name)	Specifies the display name of the hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
fld (Field)	Specifies the index of the field that appears on the page or filter report area of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema int datatype.
hier (OLAP Hierarchy Index)	Specifies the index of the OLAP hierarchy to which this item belongs. The possible values for this attribute are defined by the W3C XML Schema int datatype.
item (Item Index)	Specifies the index of the item in the PivotCache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Hierarchy Unique Name)	Specifies the unique name of the hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageField) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.63 pageFields (Page Field Items)

Represents the collection of items in the page or report filter region of the PivotTable.

[*Example*:

```
<sh:pageFields count="2">
  <sh:pageField fld="43" hier="103"
    name="[Product].[Product Categories].[All Products]" cap="All Products"/>
  <sh:pageField fld="66" hier="126"
    name="[Promotion].[Promotions].[All Promotions]" cap="All Promotions"/>
</sh:pageFields>
```

end example]

Attributes	Description
count (Page Item Count)	Specifies the number of items in the page region of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageFields) is located in §A.2. *end note*]

18.10.1.64 pageItem (Page Item)

Represents an item value for a PivotTable page.

Attributes	Description
name (Page Item Name)	Specifies the name of this page item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageItem) is located in §A.2. *end note*]

18.10.1.65 pages (Page Item Values)

Represents the collection of page item values for each page field.

Attributes	Description
count (Page Item String Count)	Specifies the number of page item strings in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pages) is located in §A.2. *end note*]

18.10.1.66 pivotAreas (Pivot Areas)

Represents the collection of pivot areas that comprise the PivotTable location.

[*Example*:

```
<sh:pivotAreas count="1">
  <sh:pivotArea field="2" dataOnly="0" outline="0"/>
</sh:pivotAreas>
```

end example]

Attributes	Description
count (Pivot Area Count)	Specifies the number of PivotAreas for the PivotTable location. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotAreas) is located in §A.2. *end note*]

18.10.1.67 pivotCacheDefinition (PivotCache Definition)

Represents the pivotCacheDefinition part. This part defines each field in the source data, including the name, the string resources of the instance data (for shared items), and information about the type of data that appears in the field.

[*Example*:

```
<pivotCacheDefinition xmlns="…" xmlns:r="…" r:id="rId1" refreshedBy="AnonUser"
  refreshedDateIso="2006-05-22T10:07:16Z" createdVersion="3"
refreshedVersion="3"
  minRefreshableVersion="3" recordCount="182">
  <cacheSource type="worksheet">
    <worksheetSource name="Table1"/>
  </cacheSource>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
  <cacheFields count="28">
    <cacheField name="Customer Name" numFmtId="0">
    <cacheField name="Postal Code" numFmtId="0">
      <sharedItems/>
    </cacheField>
    <cacheField name="Product Category" numFmtId="0">
      <sharedItems count="1">
        <s v="Bikes"/>
      </sharedItems>
    </cacheField>
    <cacheField name="Year" numFmtId="0">
      <sharedItems count="1">
        <s v="2001"/>
      </sharedItems>
    </cacheField>
    <cacheField name="Quarter" numFmtId="0">
      <sharedItems containsSemiMixedTypes="0" containsString="0"
        containsNumber="1" containsInteger="1" minValue="3" maxValue="3"
        count="1">
        <n v="3"/>
      </sharedItems>
    </cacheField>
  </cacheFields>
</pivotCacheDefinition>
```

end example]

Attributes	Description
backgroundQuery (Background Query)	Specifies a boolean value that indicates whether the application should query and retrieve records asynchronously from the cache. A value of `1` or `true` indicates the application will retrieve records asynchronously from the cache. A value of `0` or `false` indicates the application will retrieve records synchronously. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
createdVersion (PivotCache Created Version)	Specifies the version of the application that created the cache. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
enableRefresh	Specifies a boolean value that indicates whether the end-user can refresh the cache. This

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Enable PivotCache Refresh)	attribute depends on whether the application exposes a method for allowing end-users control over refreshing the cache via the user interface. A value of `1` or `true` indicates the end-user can refresh the cache. A value of `0` or `false` indicates the end-user cannot refresh the cache. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the unique identifier that corresponds to the related `pivotCacheRecords` part. See (§18.10.1.68) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
invalid (Invalid Cache)	Specifies a boolean value that indicates whether the cache needs to be refreshed. A value of `1` or `true` indicates the cache needs to be refreshed. A value of `0` or `false` indicates the cache does not need to be refreshed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minRefreshableVersion (Minimum Version Required for Refresh)	Specifies the earliest version of the application that is required to refresh the cache. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
missingItemsLimit (Missing Items Limit)	Specifies the number of unused items to allow before discarding unused items. This attribute is application-dependent. The application shall specify a threshold for unused items. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
optimizeMemory (Optimize Cache for Memory)	Specifies a boolean value that indicates whether the application will apply optimizations to the cache to reduce memory usage. This attribute is application-dependent. This application shall define its own cache optimization methods. The application shall also decide whether to expose cache optimization status via the user interface or an object model. A value of `1` or `true` indicates the application will apply optimizations to the cache.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates the application will not apply optimizations to the cache. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
recordCount (PivotCache Record Count)	Specifies the number of records in the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
refreshedBy (Last Refreshed By)	Specifies the name of the end-user who last refreshed the cache. This attribute is application-dependent and is specified by applications that track and store the identity of the current user. This attribute also depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
refreshedDateIso (PivotCache Last Refreshed Date ISO)	Specifies the date when the cache was last refreshed. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
refreshedVersion (PivotCache Last Refreshed Version)	Specifies the version of the application that last refreshed the cache. This attribute depends on whether the application exposes mechanisms via the user interface whereby the end-user can refresh the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
refreshOnLoad (Refresh On Load)	Specifies a boolean value that indicates whether the application will refresh the cache when the workbook has been opened. A value of `1` or `true` indicates that application will refresh the cache when the workbook is loaded. A value of `0` or `false` indicates the application will not automatically refresh cached data. The end user shall trigger refresh of the cache manually via the application user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saveData (Save Pivot Records)	Specifies a boolean value that indicates whether the pivot records are saved with the cache. A value of `1` or `true` indicates pivot records are saved in the cache. A value of `0` or `false` indicates are not saved in the cache.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
supportAdvancedDrill (Supports Attribute Drilldown)	Specifies whether the cache's data source supports attribute drilldown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
supportSubquery (Supports Subqueries)	Specifies whether the cache's data source supports subqueries. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tupleCache (Stores Cache for OLAP Functions)	Specifies a boolean value that indicates whether the PivotCache is used store information for OLAP sheet data functions. A value of `1` or `true` indicates information about OLAP sheet data functions are stored in the cache. A value of `0` or `false` indicates the PivotCache does not contain information about OLAP sheet data functions. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
upgradeOnRefresh (Upgrade PivotCache on Refresh)	Specifies a boolean value that indicates whether the cache is scheduled for version upgrade. This attribute depends on whether the application exposes mechanisms via the user interface whereby the cache might be upgraded. A value of `1` or `true` indicates the cache is scheduled for upgrade. A value of `0` or `false` indicates the cache is not scheduled for upgrade. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotCacheDefinition) is located in §A.2. *end note*]

18.10.1.68 pivotCacheRecords (PivotCache Records)

Represents the collection of records in the PivotCache. This part stores the underlying source data that the PivotTable aggregates.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<pivotCacheRecords xmlns="…" xmlns:r="…" count="2">
  <r>
    <x v="0"/>
    <s v="Pacific"/>
    <x v="0"/>
    <s v="Australia"/>
    <x v="0"/>
    <x v="0"/>
    <s v="3550"/>
    <x v="0"/>
    <x v="0"/>
    <s v="Road-150 Red, 62"/>
    <s v="This bike is ridden by race winners. Developed with the Adventure
      Works Cycles professional race team, it has a extremely light
      heat-treated aluminum frame, and steering that allows precision
      control."/>
    <s v="No Discount"/>
…
    <n v="89.456800000000001"/>
  </r>
…
```

end example]

Attributes	Description
count (PivotCache Records Count)	Specifies the number of records in the cache. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotCacheRecords) is located in §A.2. *end note*]

18.10.1.69 pivotField (PivotTable Field)

Represents a single field in the PivotTable. This element contains information about the field, including the collection of items in the field.

[*Example*:

```
<pivotField axis="axisRow" allDrilled="1" showAll="0" measureFilter="1"
  sortType="descending">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <items count="8">
      <item s="1" c="1" x="0"/>
      <item s="1" c="1" x="1"/>
      <item c="1" x="2"/>
      <item c="1" x="3"/>
      <item c="1" x="4"/>
      <item c="1" x="5"/>
      <item c="1" x="6"/>
      <item t="default"/>
    </items>
    <autoSortScope>
      <pivotArea dataOnly="0" outline="0" fieldPosition="0">
        <references count="2">
          <reference field="4294967294" count="1" selected="0">
            <x v="0"/>
          </reference>
          <reference field="25" count="1" selected="0">
            <x v="0"/>
          </reference>
        </references>
      </pivotArea>
    </autoSortScope>
  </pivotField>
```

end example]

Attributes	Description
allDrilled (All Items Expanded)	Specifies a boolean value that indicates whether all items in the field are expanded. Applies only to OLAP PivotTables. A value of `1` or `true` indicates all items in the field are expanded. A value of `0` or `false` indicates all items are not expanded. However some items might be expanded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoShow (Auto Show)	Specifies a boolean value that indicates whether an "AutoShow" filter is applied to this field. This attribute depends on the implementation of filtering in the application. A value of `1` or `true` indicates an "AutoShow" filter is applied to the field. A value of `0` or `false` indicates an "AutoShow" filter is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
avgSubtotal (Average)	Specifies a boolean value that indicates whether to apply the 'Average' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the subtotal for this field is 'Average.' A value of `0` or `false` indicates a different aggregation function is applied to the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
axis (Axis)	Specifies the region of the PivotTable that this field is displayed. The possible values for this attribute are defined by the ST_Axis simple type (§18.18.1).
compact (Compact)	Specifies a boolean value that indicates whether the application will display fields compactly in the sheet on which this PivotTable resides. A value of `1` or `true` indicates the next field should be displayed in the same column of the sheet. A value of `0` or `false` indicates each pivot field will display in its own column in the sheet. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
countASubtotal (CountA)	Specifies a boolean value that indicates whether to apply the 'countA' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the subtotal for this field is 'countA.' A value of `0` or `false` indicates a different aggregation function is applied to the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
countSubtotal (Count)	Specifies a boolean value that indicates whether to apply the 'count' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the subtotal for this field is 'count.' A value of `0` or `false` indicates a different aggregation vfunction is applied to the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
dataField (Data Field)	Specifies a boolean value that indicates whether this field appears in the data region of the PivotTable. A value of `1` or `true` indicates this field appears in the data region of the PivotTable. A value of `0` or `false` indicates this field appears in another region of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataSourceSort (Data Source Sort)	Specifies a boolean value that indicates whether sort is applied to this field in the data source. A value of `1` or `true` indicates this field is sorted in the data source. A value of `0` or `false` indicates this field is not sorted in the data source. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultAttributeDrillState (Drill State)	Specifies a boolean value that indicates the drill state of the attribute hierarchy in an OLAP-based PivotTable. A value of `1` or `true` indicates the attribute hierarchy is expanded. A value of `0` or `false` indicates the attribute hierarchy is collapsed. This attribute is designed to allow the application to issue more optimized queries when all items of each field have the same drill state. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultSubtotal (Show Default Subtotal)	Specifies a boolean value that indicates whether the default subtotal aggregation function is displayed for this field. A value of `1` or `true` indicates the default subtotal aggregation function is displayed for this field. A value of `0` or `false` indicates the default aggregation function is not displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragOff (Drag Off)	Specifies a boolean value that indicates whether the field can be removed from the PivotTable. A value of `1` or `true` indicates the field can be removed from the PivotTable.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates the field cannot be removed from the PivotTable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToCol (Drag To Column)	Specifies a boolean value that indicates whether the field can be dragged to the column axis. A value of `1` or `true` indicates the field can be dragged to the column axis. A value of `0` or `false` indicates the field cannot be dragged to the column axis. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToData (Field Can Drag to Data)	Specifies a boolean value that indicates whether the field can be dragged to the data region. A value of `1` or `true` indicates the field can be dragged to the data region. A value of `0` or `false` indicates the field cannot be dragged to the data region. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToPage (Drag Field to Page)	Specifies a boolean value that indicates whether the field can be dragged to the page region. A value of `1` or `true` indicates the field can be dragged to the page region. A value of `0` or `false` indicates the field cannot be dragged to the page region. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToRow (Drag To Row)	Specifies a boolean value that indicates whether the field can be dragged to the row axis. A value of `1` or `true` indicates the field can be dragged to the row axis. A value of `0` or `false` indicates the field cannot be dragged to the row axis. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hiddenLevel (Hidden Level)	Specifies a boolean value that indicates whether there is a hidden level in the PivotTable. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates the OLAP PivotTable contains a hidden level.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates the OLAP PivotTable does not contain any hidden levels. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hideNewItems (Hide New Items)	Specifies a boolean value that indicates whether new items that appear after a refresh should be hidden by default. A value of `1` or `true` indicates that items that appear after a refresh should be hidden by default. A value of `0` or `false` indicates that items that appear after a refresh should be shown by default. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
includeNewItemsInFilter (Inclusive Manual Filter)	Specifies a boolean value that indicates whether manual filter is in inclusive mode. A value of `1` or `true` indicates the manual filter is inclusive. A value of `0` or `false` indicates the manual filter is not inclusive. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
insertBlankRow (Insert Blank Row)	Specifies a boolean value that indicates whether to insert a blank row after each item. A value of `1` or `true` indicates that a blank row is inserted after each item. A value of `0` or `false` indicates no additional rows are inserted after each item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
insertPageBreak (Insert Item Page Break)	Specifies a boolean value that indicates whether to insert a page break after each item. A value of `1` or `true` indicates that a page break is inserted after each item. A value of `0` or `false` indicates no page breaks are inserted after items. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
itemPageCount (Items Per Page Count)	Specifies the number of items showed per page in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
maxSubtotal (Max Subtotal)	Specifies a boolean value that indicates whether to apply the 'max' aggregation function in the subtotal of this field. A value of `1` or `true` indicates that the 'max' aggregation function is applied in the subtotal for this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
measureFilter (Measure Filter)	Specifies a boolean value that indicates whether field has a measure based filter. A value of `1` or `true` indicates the field has a measure-based filter. A value of `0` or `false` indicates does not have a measure-based filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minSubtotal (Min Subtotal)	Specifies a boolean value that indicates whether to apply the 'min' aggregation function in the subtotal of this field. A value of `1` or `true` indicates that the 'min' aggregation function is applied in the subtotal for this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
multipleItemSelectionAllowed (Multiple Field Filters)	Specifies a boolean value that indicates whether the field can have multiple items selected in the page field. A value of `1` or `true` indicates the PivotTable can have multiple items selected in the page field. A value of `0` or `false` indicates the PivotTable cannot have multiple items selected in the page field. This attribute depends on the application support for selecting multiple items in page fields. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Field Name)	Specifies the name of the field. The possible values for this attribute are defined by the ST_Xstring simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
nonAutoSortDefault (Auto Sort)	Specifies a boolean value that indicates whether sort operation that is applied to field should be AutoSort operation or simple data sort operation. A value of `1` or `true` indicates that an AutoSort operation is applied to the field. A value of `0` or `false` indicates a simple data sort operation is applied to the field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
numFmtId (Number Format Id)	Specifies the identifier of the number format to apply to this field. Number formats are written to the styles part. See the Styles section (§18.8) for more information on number formats. Formatting information provided by cell table and by PivotTable need not agree. If the two formats differ, the cell-level formatting takes precedence. If you change the layout the PivotTable, the PivotTable formatting will then take precedence. The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
outline (Outline Items)	Specifies a boolean value that indicates whether the items in this field should be shown in Outline form. A value of `1` or `true` indicates the items in this field is shown in Outline form. A value of `0` or `false` indicates the items in this field will not be shown in Outline form. This attribute depends on the application support for displaying items in Outline form. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
productSubtotal (Product Subtotal)	Specifies a boolean value that indicates whether to apply 'product' aggregation function in the subtotal of this field. A value of `1` or `true` indicates that the 'product' aggregation function is applied in the subtotal for this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rankBy (Auto Show Rank By)	Specifies the index of the data field by which AutoShow will rank. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
serverField (Server-based Page Field)	Specifies a boolean value that indicates whether this is a server-based page field. A value of `1` or `true` indicates this is a server-based page field. A value of `0` or `false` indicates this is a local page field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showAll (Show All Items)	Specifies a boolean value that indicates whether to show all items for this field. A value of `1` or `true` indicates that all items be shown. A value of `0` or `false` indicates items be shown according to user specified criteria. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showDropDowns (Show PivotField Header Drop Downs)	Specifies a boolean value that indicates whether to hide drop down buttons on PivotField headers. This attribute depends on the application implementation for filtering in the user interface. A value of `1` or `true` indicates the application will display some mechanism for selecting and applying filters – [*Example*: A dropdown menu *end example*] – in the user interface. A value of `0` or `false` indicates for mechanism for applying a filter is displayed in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showPropAsCaption (Show As Caption)	Specifies a boolean value that indicates whether to show the property as a member caption. A value of `1` or `true` indicates the property is shown as a member caption. A value of `0` or `false` indicates the property will not be shown as a member caption. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showPropCell (Show Member Property in Cell)	Specifies a boolean value that indicates whether to show the member property value in a PivotTable cell. A value of `1` or `true` indicates the property value is shown in a PivotTable cell. A value of `0` or `false` indicates the property value will not be shown in a PivotTable cell.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showPropTip (Show Member Property ToolTip)	Specifies a boolean value that indicates whether to show the member property value in a tooltip on the appropriate PivotTable cells. A value of `1` or `true` indicates the property value is shown in a tooltip in the user interface. A value of `0` or `false` indicates the property will not be shown in a tooltip. This attribute depends on whether the application employs tooltips or similar mechanism in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sortType (Auto Sort Type)	Specifies the type of sort that is applied to this field. The possible values for this attribute are defined by the ST_FieldSortType simple type (§18.18.28).
stdDevPSubtotal (StdDevP Subtotal)	Specifies a boolean value that indicates whether to apply the 'stdDevP' aggregation function in the subtotal of this field. A value of `1` or `true` indicates that the 'stdDevP' aggregation function is applied in the subtotal for this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
stdDevSubtotal (StdDev Subtotal)	Specifies a boolean value that indicates whether to use 'stdDev' in the subtotal of this field. A value of `1` or `true` indicates that the 'stdDev' aggregation function is applied in the subtotal for this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal for this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
subtotalCaption (Custom Subtotal Caption)	Specifies the custom text that is displayed for the subtotals label. The possible values for this attribute are defined by the ST_Xstring simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
subtotalTop (Subtotals At Top)	Specifies a boolean value that indicates whether to display subtotals at the top of the group. Applies only when `Outline` its true. A value of `1` or `true` indicates a subtotal is display at the top of the group. A value of `0` or `false` indicates subtotal will not be displayed at the top of the group. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sumSubtotal (Sum Subtotal)	Specifies a boolean value that indicates whether apply the 'sum' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the 'sum' aggregation function is applied in the subtotal of this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal of this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
topAutoShow (Top Auto Show)	Specifies a boolean value that indicates whether an AutoShow filter applied to this field is set to show the top ranked values. A value of `1` or `true` indicates whether an AutoShow filter will show top values for this field. A value of `0` or `false` indicates bottom ranked values are shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
uniqueMemberProperty (Unique Member Property)	Specifies the unique name of the member property to be used as a caption for the field and field items. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
varPSubtotal (VarP Subtotal)	Specifies a boolean value that indicates whether to apply the 'varP' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the 'varP' aggregation function is applied in the subtotal of this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal of this field.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
varSubtotal (Variance Subtotal)	Specifies a boolean value that indicates whether to apply the 'variance' aggregation function in the subtotal of this field. A value of `1` or `true` indicates the 'variance' aggregation function is applied in the subtotal of this field. A value of `0` or `false` indicates another aggregation function is applied in the subtotal of this field. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotField) is located in §A.2. *end note*]

18.10.1.70 pivotFields (PivotTable Fields)

Represents the collection of fields that appear on the PivotTable.

[*Example*:

```
<pivotFields count="28">
  <pivotField showAll="0" includeNewItemsInFilter="1">
    <items count="8">
      <item x="66"/>
      <item x="133"/>
      <item x="74"/>
      <item x="27"/>
      <item x="118"/>
      <item x="63"/>
      <item x="141"/>
      <item t="default"/>
    </items>
  </pivotField>
  <pivotField showAll="0" includeNewItemsInFilter="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <pivotField axis="axisPage" showAll="0" includeNewItemsInFilter="1">
      <items count="2">
        <item x="0"/>
        <item t="default"/>
      </items>
    </pivotField>
  <pivotField showAll="0" includeNewItemsInFilter="1"/>
```

end example]

Attributes	Description
count (Field Count)	Specifies the number of fields in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFields) is located in §A.2. *end note*]

18.10.1.71 pivotHierarchies (PivotTable OLAP Hierarchies)

Represents the collection of OLAP hierarchies associated with the PivotTable.

[*Example*:

```
  <sh:pivotHierarchies count="3">
    <sh:pivotHierarchy dragToRow="0" dragToCol="0" dragToPage="0" dragToData="1"/>
    <sh:pivotHierarchy dragToRow="0" dragToCol="0" dragToPage="0" dragToData="1"/>
    <sh:pivotHierarchy dragToRow="0" dragToCol="0" dragToPage="0" dragToData="1"/>
    <sh:pivotHierarchy dragToRow="0" dragToCol="0" dragToPage="0" dragToData="1"/>
  </sh:pivotHierarchies>
```

end example]

Attributes	Description
count (OLAP Hierarchy Count)	Specifies the number of OLAP hierarchies in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotHierarchies) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.10.1.72 pivotHierarchy (OLAP Hierarchy)

Represents a OLAP hierarchy associated with the PivotTable. A hierarchy is a hierarchical representation of related OLAP dimensions. Hierarchies are defined on the OLAP server and cannot be changed in the PivotTable. [*Example*: Hierarchy "A" might be defined as follows:

Level 1 Country/Region

Level 2 State\Provence

Level 3 City

end example]

[*Example*:

```
<sh:pivotHierarchy dragToRow="0" dragToCol="0" dragToPage="0" dragToData="1"/>
```

end example]

Attributes	Description
caption (Hierarchy Caption)	Specifies the user defined caption of the hierarchy. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dragOff (Drag Off)	Specifies a boolean value that indicates whether the user is allowed to remove this hierarchy from the PivotTable. A value of `1` or `true` indicates the user can remove this hierarchy from the PivotTable. A value of `0` or `false` indicates the user cannot remove the hierarchy from the PivotTable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToCol (Drag To Column)	Specifies a boolean value that indicates whether the user is allowed to put this hierarchy into the column area of the PivotTable. A value of `1` or `true` indicates the user can put this hierarchy into the column area of the PivotTable. A value of `0` or `false` indicates the user cannot remove this hierarchy. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToData (Drag To Data)	Specifies a boolean value that indicates whether the user is allowed to put this hierarchy into the data area of the view.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `1` or `true` indicates A value of `0` or `false` indicates The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToPage (Drag to Page)	Specifies a boolean value that indicates whether the user is allowed to put this hierarchy into the page area of the PivotTable. A value of `1` or `true` indicates the user can put this hierarchy into the page area of the PivotTable. A value of `0` or `false` indicates cannot put this hierarchy into the page area. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dragToRow (Drag To Row)	Specifies a boolean value that indicates whether the user is allowed to put this hierarchy into the row area of the PivotTable. A value of `1` or `true` indicates the user can put this hierarchy into the row area of the PivotTable. A value of `0` or `false` indicates cannot put this hierarchy into the row area. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
includeNewItemsInFilter (Inclusive Manual Filter)	Specifies a boolean value that indicates whether the application will show only the items the user has selected. A value of `1` or `true` indicates the application will show only items the user has selected; all other items are hidden. A value of `0` or `false` indicates the application will show all items. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
multipleItemSelectionAllowed (Multiple Field Filters)	Specifies a boolean value that indicates whether the user can select multiple members when the hierarchy is in the page field area of the view. A value of `1` or `true` indicates the user can select multiple members. A value of `0` or `false` indicates the user cannot select multiple members. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
outline (Outline New Levels)	Specifies a boolean value that indicates whether new levels added to the PivotTable are shown in Outline mode. A value of `1` or `true` indicates new levels are shown in Outline mode. A value of `0` or `false` indicates new items are not shown in Outline mode. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showInFieldList (Show In Field List)	Specifies a boolean value that indicates whether this hierarchy is omitted from the field list. This attribute depends on how the application exposes a list of fields for PivotTables in the user interface. A value of `1` or `true` indicates this hierarchy is show in the field list or similar mechanism in the user interface. A value of `0` or `false` indicates is not shown in the field list. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
subtotalTop (New Levels Subtotals At Top)	Specifies a boolean value that indicates whether new levels added to the view will show their subtotals at the top. A value of `1` or `true` indicates new levels added to the view show their subtotals at the top. A value of `0` or `false` indicates new levels added to the view show their subtotals at the bottom. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotHierarchy) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.10.1.73 pivotTableDefinition (PivotTable Definition)

Represents the PivotTable root element for non-null PivotTables. There exists one pivotTableDefinition for each PivotTableDefinition part. The PivotTable definition encompasses the following information:

Structure

- Top-level attributes

 ©ISO/IEC 2016 – All rights reserved

- Location information
- Collection of fields
- Fields on the row axis
- Items on the row axis (specific values)
- Fields on the column axis
- Items on the column axis (specific values)
- Fields on the report filter region
- Fields in the values region
- Style information

Outline of the XML for a pivotTableDefinition

```
<pivotTableDefinition>
  <location/>
  <pivotFields/>
  <rowFields/>
  <rowItems/>
  <colFields/>
  <colItems/>
  <pageFields/>
  <dataFields/>
  <conditionalFormats/>
  <pivotTableStyleInfo/>
</pivotTableDefinition>
```

Layout

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The reference specified in the ref attribute on the location element specifies the location of the PivotTable body. The data area, row, column, and data fields and value items are located in this area. More specifically, the row fields begin below the A1-most cell in the reference, and the column fields begin adjacent to that cell, in the same row, extending out into the PivotTable body away from column A. *[Note:* How far below or across the field labels begin are dependent upon how many row, column, and data fields are shown in the PivotTable. More detail is provided below. *end note]*

[*Note*: All layout discussion and examples are given for outline mode layout. There are two additional layout modes: compact and tabular. See Other layout modes below for a discussion of how those differ from outline mode. *end note*]

©ISO/IEC 2016 – All rights reserved

When encountering sheet boundaries, the PivotTable is truncated rather than wrapped, and as much as possible shall be shown.

The graphics given in this section are meant to illustrate layout only, and do not require implementation of any implied controls, like dropdowns or expand/collapse functionality.

Page Field Layout

	A	B	C	D
1				
2		SSN	(All)	
3				
4		State	City	Sum of Amount
5		CA		195.51
6			San Diego	195.51
7		OR		54.97
8			Portland	12.54
9			Tillamook	42.43
10		WA		244.12
11			Seattle	96.72
12			Tacoma	79.83
13			Everett	67.57
14		Grand Total		494.6

In the above picture, SSN is a page field, State and City are row fields, and Amount is a data field. There are no column fields.

Page fields allow you to filter the entire PivotTable report to display data for a single item or all items.

The page field area always ends (vertically) so that there is always 1 row of space between the page field area and the top row of the PivotTable body, and always begins (horizontally) in the same column as the A1-most column of the PivotTable body. Each page field occupies two cells: the A1-most for displaying the field name, and the next cell over for displaying the selected item values. [*Example*: (see above picture) If the top row in the PivotTable body reference is row 4, then page field layout ends (vertically) in row 2, and if the A1-most column of the PivotTable body is column B, then page field layout begins (horizontally) in column B. *end example*]

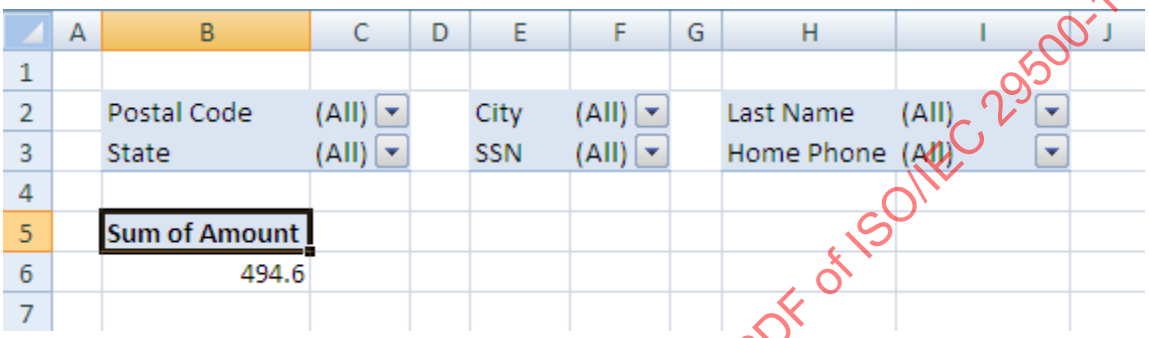
Aside from the number of fields in the page field area, there are two attributes of pivotTableDefinition that affect page field layout: pageOverThenDown and pageWrap. pageOverThenDown = 1 specifies that when there is more than 1 page field, lay them out horizontally across the sheet (extending in the direction of the PivotTable body area, away from column A) until the maximum specified in pageWrap is reached, and then begin a new row. If the pageWrap value is high and there are many page fields, then it is possible (and allowed) for page fields to extend beyond the edge of the PivotTable body. When laying out page fields in the same row (side by side), each shall be separated by a single column. However, multiple rows of page fields are not separated by single rows between them. pageOverThenDown = 0 specifies that when there is more than 1 page

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

field, lay them out vertically down the sheet (always keeping 1 row of space between the PivotTable body and page field area) until the maximum specified in pageWrap is reached, and then begin a new column. Again, for multiple page fields, if they shall occupy more than 1 column, then each column of page fields is separated by a single column, and multiple rows of page fields are not separated by single rows between them.

[*Example:* This example shows a PivotTable body occupying B5:B6 and 6 page fields in the page field area, where pageOverThenDown = `0` and pageWrap = 2. This means that the first column of the page field area contains 2 page fields, and then, because the pageWrap value only allows 2 page fields per column, a new column of page fields is started, and so on until all 6 page fields are shown.



The order of assignment of position within page field layout for this example is:

- Postal Code
- State
- City
- SSN
- Last Name
- Home Phone

Aside from the 6 page fields, the only other field in this PivotTable example is a data field called Amount.

end example]

[*Note:* When the user gestures to add a page field and there are not enough free cells above the PivotTable body area to allow for page fields to be added, the application must determine the best response. The application may decide to shift the PivotTable down some number of rows to make room, or overwrite existing data or features that might be above the PivotTable, or simply block the user gesture completely. In any result, however, the application should adhere to the layout principles given above. *end note*]

Row Field Layout

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	A	B	C	D
1				
2		SSN	(All)	
3				
4		State	City	Sum of Amount
5		CA		195.51
6			San Diego	195.51
7		OR		54.97
8			Portland	12.54
9			Tillamook	42.43
10		WA		244.12
11			Seattle	96.72
12			Tacoma	79.83
13			Everett	67.57
14		Grand Total		494.6

The **State** and **City** fields are row fields, **SSN** is a page field, there are no column fields, and **Amount** is a data field.

Row fields provide for and specify how the data is summarized, grouped, and viewed as rows in the PivotTable.

The row field area always begins in the A1-most column of the PivotTable body area. The layout of page fields does not affect the layout of row fields.

Row Field Layout - 1 Row Field and 0 Column Fields

When there is only 1 row field and 0 column fields,

the first row field is located in the A1-most cell of the PivotTable body, and
the values for that field are expressed in the cells directly under that row field, in the same column.

[*Example:*

State	Sum of Amount
CA	195.51
OR	54.97
WA	244.12
Grand Total	494.6

In this example, there are no page fields, no column fields, **State** is a row field, and **Amount** is a data field. *end example*]

Row Field Layout - 2 or More Row Fields and 0 Column Fields

When there are 2 or more row fields and 0 column fields to be displayed,

- the row field labels are located adjacent to each other and in the same row as the first row field label
- Each corresponding set of values for the row field in question are located in the cells under that row field (same column)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- Innermost row field values (the ones closest to the data summary area) are grouped and organized by values in the next outer row field, in the following fashion: starting with the outermost row field, the first value is listed. For the next innermost row field, starting on the next row and over one column (toward the data summary area), the value list for that field begins. If that is the innermost row field, all values are listed for that row field, and then moving down a row and back to the outer column, the next value for the outermost row field is listed. If there are more inner row fields, the same layout rules apply until the innermost row field is reached.
- In this case of 0 column fields, only the top row of the PivotTable body is used for row field labels.

[*Example:*

	A	B	C	D	E
1					
2					
3					
4		Postal Co	State	City	Sum of Amount
5		09999			54.97
6			OR		54.97
7				Portland	12.54
8				Tillamook	42.43
9		12345			195.51
10			CA		195.51
11				San Diego	195.51
12		456789			244.12
13			WA		244.12
14				Seattle	96.72
15				Tacoma	79.83
16				Everett	67.57
17		Grand Total			494.6

In this example **Postal Code**, **State**, and **City** are row fields and **Amount** is a data field. There are no page fields and no column fields. *end example*]

Row Field Layout - 1 or More Row Fields and 1 or More Column Fields

When there are row fields and 1 or more column fields, the row fields are not located in the topmost row of the PivotTable body. Instead the row fields are located in the n+1st topmost row of the PivotTable body, where n is the number of column fields in the PivotTable.

[*Example:*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E
1					
2					
3					
4					
5		Sum of Amount	Postal Code	State	City
6			09999		
7			OR		OR Total
8		Last Name	Portland	Tillamook	
9		Cencini	12.54		12.54
10		Freehafer			
11		Giussani			
12		Hellung-Larsen			
13		Kotas		42.43	42.43
14		Neipper			
15		Sergienko			
16		Thorpe			
17		Zare			
18		Grand Total	12.54	42.43	54.97

This example shows 3 column fields in the PivotTable (**Postal Code**, **State**, and **City**), a single row field **Last Name**, and a single data field **Amount**. The PivotTable body area begins at B5 and the row field label **Last Name** is located in the 4th row of the PivotTable body area, in row 8 of the spreadsheet, cell B8. Since **Last Name** is the only row field in this example, its row field values begin and are listed directly under the label. *end example*]

Column Field Layout

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5		Sum of Amount	State	City				
6			CA	OR		WA		
7		Last Name	San Diego	Portland	Tillamook	Seattle	Tacoma	Everett
8		Cencini		12.54				
9		Freehafer				53.34		
10		Giussani					79.83	
11		Hellung-Larsen						67.57
12		Kotas			42.43			
13		Neipper	63.67					
14		Sergienko	50.69					
15		Thorpe	81.15					
16		Zare				43.38		

State and **City** are column fields, **Last Name** is a row field, and **Amount** is a data field.

Column fields provide for and specify how the data is summarized, grouped, and viewed horizontally in the PivotTable.

The layout of page fields does not affect the layout of column fields.

The column field label area is always located in the top row of the PivotTable body.

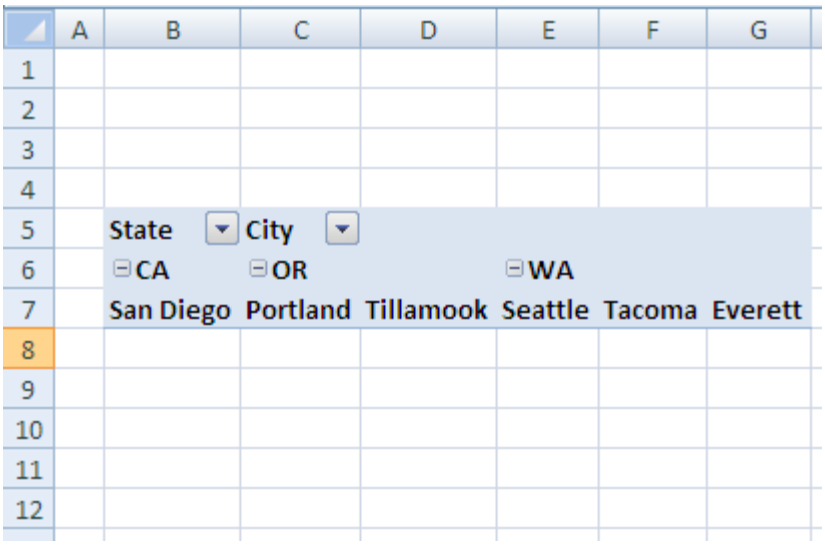
Column Field Layout - 0 Row Fields and 0 Data Fields

- When there are no row fields and no data fields, then the first column field is located in the A1-most column of the PivotTable body.
- When there are multiple column fields
- the labels are located adjacent to each other in the same row as the first column field label.
- Each corresponding set of values for each of the column fields are located in the rows directly below the column field label row and above the data area, one row of values for each column field.
- The first column field's values are located in the row directly under the column field row.
- Column field values are displayed starting directly underneath the first column field label's cell and filling adjacent cells in the same row. The second column field's values are located two rows under the column field label row, and values are again displayed starting directly underneath the first column field label's cell, filling adjacent cells in the same manner as the first set of values. The layout of column field values continues in this way until all column field values are displayed.

Inner column field values (the ones closer to the data summary area) are grouped and organized by values in the next outer column field, similarly to how row field values are grouped. [*Example:*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



	A	B	C	D	E	F	G
1							
2							
3							
4							
5		State	City				
6		CA	OR		WA		
7		San Diego	Portland	Tillamook	Seattle	Tacoma	Everett
8							
9							
10							
11							
12							

In this example, **State** and **City** are column fields, and there are no row fields, no page fields, and no data fields. *end example*]

Column Field Layout – 1 or More Column Fields and 1 or More Row Fields

When there are 1 or more column fields and 1 or more row fields in the PivotTable, then:

- First, row fields are displayed according to the row field layout described earlier
- The first column field label is located in the top row of the PivotTable body area, and adjacent to any row field labels that are displayed.
- Multiple column fields shall be displayed as described earlier

[*Example:* In this example, **State** and **City** are column fields, **Amount** is a data field, and **Last Name** is a row field.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5		Sum of Amount	State	City				
6			CA	OR		WA		
7		Last Name	San Diego	Portland	Tillamook	Seattle	Tacoma	Everett
8		Cencini		12.54				
9		Freehafer				53.34		
10		Giussani					79.83	
11		Hellung-Larsen						67.57
12		Kotas			42.43			
13		Neipper	63.67					
14		Sergienko	50.69					
15		Thorpe	81.15					
16		Zare				43.38		

end example]

Data Field Layout

	A	B	C
1			
2			
3			
4			
5		Last Name	Sum of Amount
6		Cencini	12.54
7		Freehafer	53.34
8		Giussani	79.83
9		Hellung-Larsen	67.57
10		Kotas	42.43
11		Neipper	63.67
12		Sergienko	50.69
13		Thorpe	81.15
14		Zare	43.38

Last Name is a row field, Sum of Amount is a data field label, and the data underneath Sum of Amount are the summarized data values.

Data fields specify which fields are summarized in the PivotTable report.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The summarized data always appears below the column field and value area, and any row field values are closer to column A than any of the summarized data. When there are no row fields and no column fields, the summarized data is located directly under the A1-most cell of the PivotTable body. Each cell in the summarized data area represents an aggregation of a set of records. The set of records that a particular cell is summarizing is determined by looking at the row field value(s) and column field value(s) that intersect on that particular cell, and then determining which records in the source data contain all of those row and column field values.

Data Field Layout - 0 Row Fields and 0 Column Fields and 1 Data Field

When there are no row fields and no column fields and only 1 data field being summarized, the data field label is located in the A1-most cell of the PivotTable body.

[*Example:*

	A	B
1		
2		**Sum of Amount**
3		494.6

In this example there is only 1 field in the PivotTable, a data field **Amount**. *end example*]

Data Field Layout - More Than 1 Data Field

When there is more than 1 data field being summarized,

- An additional field (in these examples labeled “Values”, but the label can be specified by the user) is added to the field list, located as either a row field label or a column field label (depending on user choice and behaviour as specified by the dataOnRows and dataPosition attributes), and
- each data field being summarized is displayed either in the row area (when the additional field is a row field) as if it were an item value of that row field (see row field layout description above), or in the column area (when the additional field is a column field) as if it were an item value of that column field (see column field layout description above).

[*Example:*

	A	B	C
1			
2		**Values**	
3		**Sum of Amount**	**Sum of Tax**
4		494.6	28.71

In this example there are 2 data fields **Amount** and **Tax**. There are no page fields, no column fields, no row fields, and the additional field labeled Values is placed on the column area.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	A	B	C
1			
2		**Values**	
3		Sum of Amount	494.6
4		Sum of Tax	28.71

Above is the same PivotTable, with the Values field placed on the row area.

end example]

Data Field Layout - 0 Row Fields, 1 or More Column Fields, and 1 Data Field

When there are no row fields, 1 or more column fields, and only 1 data field being summarized, the data field label is located in the A1-most column of the PivotTable body, directly under the column field area.

[*Example:*

	A	B	C	D	E
1					
2			**State**		
3			**CA**	**OR**	**WA**
4		Sum of Amount	195.51	54.97	244.12

In this example there is 1 column field **State** and 1 data field **Amount**. There are no row fields or page fields. *end example*]

Data Field Layout - 0 Column Fields, 1 or More Row Fields, and 1 Data Field

When there are no column fields, 1 or more row fields, and only 1 data field being summarized, the data field label is located in the same row as the row field labels, above the data summary area.

[*Example:*

	A	B	C
1			
2		**State**	**Sum of Amount**
3		CA	195.51
4		OR	54.97
5		WA	244.12

In this example there is 1 data field **Amount** and 1 row field **State**. There are no column fields or page fields. *end example*]

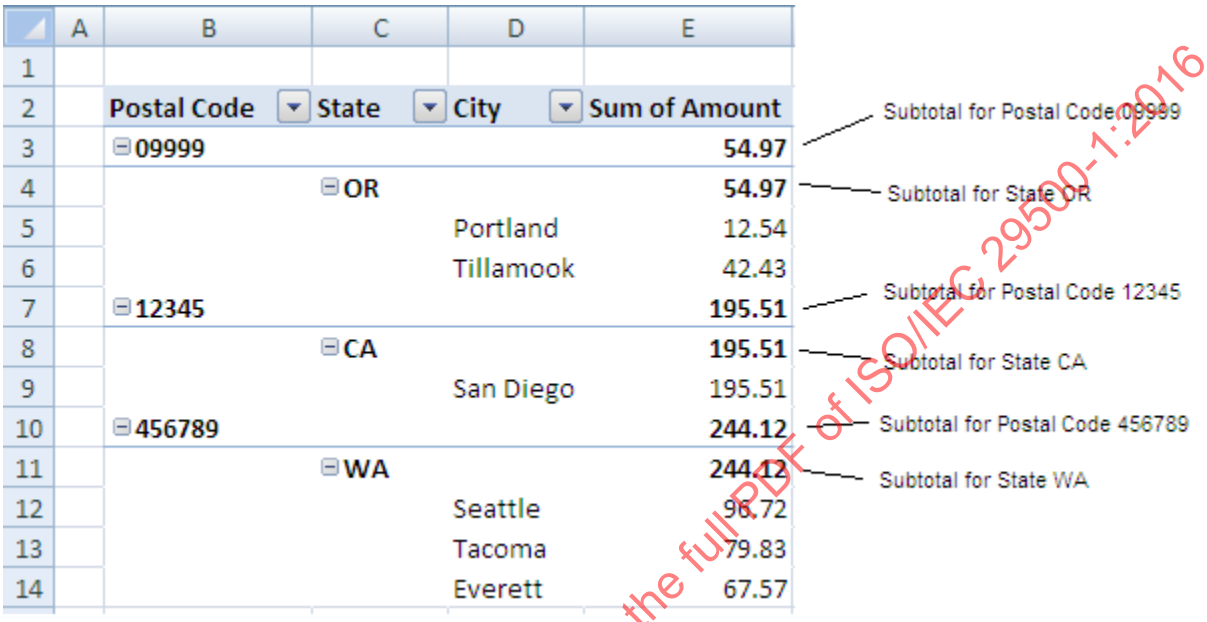
Subtotal and grand total layout

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

If subtotals are *on*, the values for row subtotals are placed at either the top of each group of data being summarized or at the bottom of each group, as indicated by the subtotalTop attribute value on the pivotField element. Row subtotal values appear in the same column as the data being subtotalled. If placed at the top of the group, then the subtotal value for the group appears in the row above the group of values, in the same row as the group's parent row field value. When there is only a single row field, no subtotal is shown.

[*Example:*



In this example, there are 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field **Amount**.

end example]

If row subtotals are placed at the bottom of each data group, then a new row is inserted directly below the data group in question, and a new row field value is inserted, in the same column as the row field in question, whose caption indicates that this row represents a subtotal value.

[*Example:*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E
1					
2		Postal Code	State	City	Sum of Amount
3		09999			
4			OR		
5				Portland	12.54
6				Tillamook	42.43
7			OR Total		54.97
8		09999 Total			54.97
9		12345			
10			CA		
11				San Diego	195.51
12			CA Total		195.51
13		12345 Total			195.51
14		456789			
15			WA		
16				Seattle	96.72
17				Tacoma	79.83
18				Everett	67.57
19			WA Total		244.12
20		456789 Total			244.12

Subtotal for State OR
Subtotal for Postal Code 09999
Subtotal for State CA
Subtotal for Postal Code 12345
Subtotal for State WA
Subtotal for Postal Code 456789

In this example, there are 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field **Amount**.

end example]

If subtotals are *on*, for column subtotals a new column is inserted directly after the data group being subtotalled. A new column field value is inserted, in the same row as the column field in question, whose caption indicates that this column represents a subtotal value. When there is only a single column field, no subtotal is shown.

[*Example:*

	A	B	C	D	E	F	G	H	I
1									
2			Postal Code	State	City				
3			09999			09999 Total	12345		12345 Total
4			OR		OR Total		CA	CA Total	
5			Portland	Tillamook			San Diego		
6		Sum of Amount	12.54	42.43	54.97	54.97	195.51	195.51	195.51

Subtotal for State OR
Subtotal for State CA
Subtotal for Postal Code 09999
Subtotal for Postal Code 12345

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

In this example, there are 3 column fields (**Postal Code**, **State**, and **City**) and a data field **Amount**. *end example*]

If row grand totals are on and there are column fields, a new column item is inserted at the very edge of the PivotTable body furthest away from column A, in the same row as the outermost column field values. The caption indicates that this is a grand total, and the values total all values across the row. When row grand totals are *on* but there are no column fields, no row grand total is shown.

[*Example:*

	A	B	C	D	E	F
1						
2			State			
3			CA	OR	WA	Grand Total
4		Sum of Amount	195.51	54.97	244.12	494.6

In this example there is 1 column field **State** and 1 data field **Amount**, and row grand totals are on.

end example]

When column grand totals are *on* and there are row fields, a new row item is inserted at the very bottom of the PivotTable body, in the same column as the outermost row field values. The caption indicates that this is a grand total, and the values total all values in the column. When column grand totals are *on* but there are no row fields, no column grand total is shown.

[*Example:*

	A	B	C
1			
2		State	Sum of Amount
3		CA	195.51
4		OR	54.97
5		WA	244.12
6		Grand Total	494.6

In this example there is 1 row field **State** and 1 data field **Amount, and column grand totals are on**.

end example]

Other Layout Modes

A PivotTable can be displayed in Compact, Outline, or Tabular form. In addition, Classic layout can be applied to any of the 3 layout forms.

Outline mode has been discussed in the above sections, and all examples are shown using outline mode with classic layout off (gridDropZones = 0).

For Compact mode, the layout differs from outline mode by:

IECNORM.COM: Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Instead of multiple row fields occupying multiple columns, the A1-most column of the PivotTable body contains all row field labels and values. A single label, “Row Labels”, is located where the first (outermost) row label is placed. When there are multiple row fields, the outermost list of values is not indented, then next inner row field values are indented (as specified in the indent attribute), and so on until each set of values for inner row fields are shown.
- Instead of multiple column fields being listed and located across a row, the first column field position is labeled “Column Labels”, and there is only this label, located in the first column field position.

[*Example:*

Outline mode:

	A	B	C	D	E	F	G
1							
2		**Sum of Amount**		**Postal Code**	**Last Name**		
3				**09999**		**12345**	
4		**State**	**City**	**Cencini**	**Kotas**	**Neipper**	**Sergienko**
5		**CA**					
6			San Diego			63.67	50.69
7		**OR**					
8			Portland	12.54			
9			Tillamook		42.43		
10		**WA**					
11			Seattle				
12			Tacoma				
13			Everett				

The above picture shows 2 column fields (**Postal Code** and **Last Name**), 1 data field (**Amount**), and 2 row fields (**State** and **City**). There are no page fields shown.

Same PivotTable in compact mode:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E	F
1						
2		Sum of Amount	Column Labels			
3			09999		12345	
4		Row Labels	Cencini	Kotas	Neipper	Sergienko
5		CA				
6		San Diego			63.67	50.69
7		OR				
8		Portland	12.54			
9		Tillamook		42.43		
10		WA				
11		Seattle				
12		Tacoma				
13		Everett				

The above picture shows all column field labels collapsed into a single label **Column Labels** and all row field labels collapsed into a single label **Row Labels**. There is 1 data field **Amount** and no page fields. *end example*]

For Tabular mode, the layout differs from outline mode by:

Instead of beginning new inner row field values on the next row down from the outer row field value parent, the first next-inner row field value is located on the same row as the parent value.
Row subtotals can only appear at the bottom of a group, not at the top

[*Example:*

Outline mode:

	A	B	C	D	E
1					
2		Postal Code	State	City	Sum of Amount
3		09999			
4			OR		
5				Portland	12.54
6				Tillamook	42.43
7		12345			
8			CA		
9				San Diego	195.51
10		456789			
11			WA		
12				Seattle	96.72
13				Tacoma	79.83
14				Everett	67.57

The above picture shows 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field, **Amount**.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Same PivotTable in tabular mode:

	A	B	C	D	E
1					
2		Postal Code	State	City	Sum of Amount
3		09999	OR	Portland	12.54
4				Tillamook	42.43
5		12345	CA	San Diego	195.51
6		456789	WA	Seattle	96.72
7				Tacoma	79.83
8				Everett	67.57

The above picture shows 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field, **Amount**.

end example]

For Classic layout, the layout differs by:

- When there are row fields, no column fields, and 1 data field, instead of displaying the data field label adjacent to and in the same row as the row field labels, the data field label is located in the A1-most cell of the PivotTable body, and the row directly under this cell contains the row field labels.
- In the exact location where the data field label is located when classic layout is off, a label titled “Total” is displayed when classic layout is on.

[*Example:*

Outline mode, classic layout off:

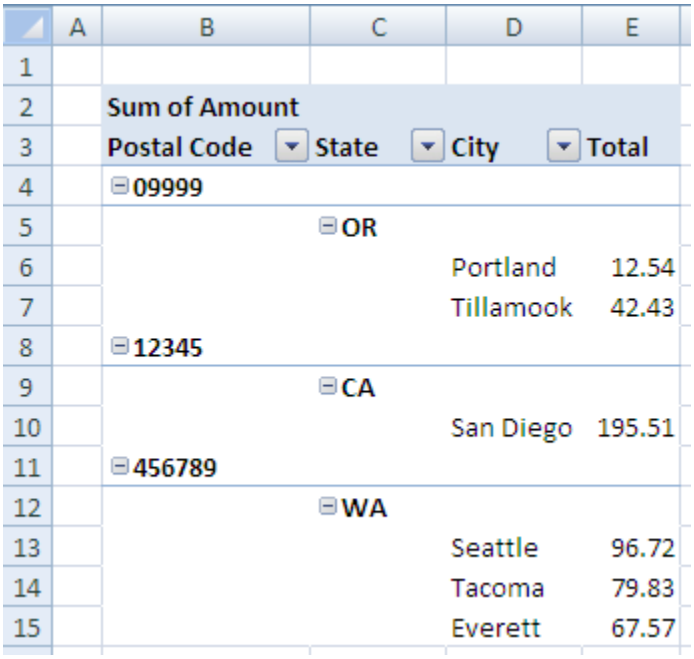
	A	B	C	D	E
1					
2		Postal Code	State	City	Sum of Amount
3		09999			
4			OR		
5				Portland	12.54
6				Tillamook	42.43
7		12345			
8			CA		
9				San Diego	195.51
10		456789			
11			WA		
12				Seattle	96.72
13				Tacoma	79.83
14				Everett	67.57

The above picture shows 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field, **Amount**.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Same PivotTable in Outline mode, classic layout applied:



	A	B	C	D	E
1					
2		Sum of Amount			
3		Postal Code	State	City	Total
4		09999			
5			OR		
6				Portland	12.54
7				Tillamook	42.43
8		12345			
9			CA		
10				San Diego	195.51
11		456789			
12			WA		
13				Seattle	96.72
14				Tacoma	79.83
15				Everett	67.57

The above picture shows 3 row fields (**Postal Code**, **State**, and **City**) and 1 data field, **Amount**.

end example]

Attributes	Description
applyAlignmentFormats (Apply Alignment Formats)	If `true` apply legacy table autoformat alignment properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyBorderFormats (Apply Border Formats)	If `true` apply legacy table autoformat border properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyFontFormats (Apply Font Formats)	If `true` apply legacy table autoformat font properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyNumberFormats (Apply Number Formats)	If `true` apply legacy table autoformat number format properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyPatternFormats (Apply Pattern Formats)	If `true` apply legacy table autoformat pattern properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
applyWidthHeightFormats (Apply Width / Height Formats)	If `true` apply legacy table autoformat width/height properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
asteriskTotals (Asterisk Totals)	Specifies a boolean value that indicates whether an asterisks should be displayed in subtotals and totals when visual totals are not used in OLAP -based PivotTables. A value of `1` or `true` indicates an asterisks are displayed in subtotals and totals for OLAP PivotTables when visual tools are not available. A value of `0` or `false` indicates an asterisk will not be displayed. This attribute depends on the implementation and availability of visual tools in the application user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoFormatId (Auto Format Id)	Identifies which legacy table autoformat to apply. Annex D contains a listing of the supported PivotTable AutoFormats, example formatting, and a sample workbook with each of those AutoFormats applied. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
cacheId (PivotCache Definition Id)	Specifies the identifier of the related PivotCache definition. This Id is listed in the pivotCaches collection in the workbook part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
chartFormat (Chart Format Id)	Specifies the next chart formatting identifier to use on the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
colGrandTotals (Grand Totals On Columns)	Specifies a boolean value that indicates whether grand totals should be displayed for the PivotTable columns. A value of `1` or `true` indicates grand totals should be displayed. A value of `0` or `false` indicates grand totals should not be displayed for PivotTable columns. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
colHeaderCaption (Column Header Caption)	Specifies the string to be displayed in column header in compact mode. This attribute depends on whether the application implements a compact mode for displaying PivotTables in the user interface. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
compact (Compact New Fields)	Specifies a boolean value that indicates whether new fields should have their `compact` flag set to true. A value of `1` or `true` indicates new fields should default to `compact` mode equal to true. A value of `0` or `false` indicates new fields should default to `compact` mode equal to false. This attribute depends on whether the application implements a compact mode in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
compactData (Compact Data)	Specifies a boolean value that indicates whether the field next to the data field in the PivotTable should be displayed in the same column of the spreadsheet The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
createdVersion (PivotCache Created Version)	Specifies the version of the application that created the cache. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
customListSort (Custom List AutoSort)	Specifies a boolean value that indicates whether the "custom lists" option is offered when sorting this PivotTable. A value of `1` or `true` indicates custom lists are offered when sorting this PivotTable. A value of `0` or `false` indicates custom lists are not offered. This attribute depends on the implementation of sorting features in the application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataCaption (Data Field Header Name)	Specifies the name of the value area field header in the PivotTable. This caption is shown when the PivotTable when two or more fields are in the values area. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dataOnRows (Data On Rows)	Specifies a boolean value that indicates whether the field representing multiple fields in the data region is located in the row area or the column area.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `1` or `true` indicates that this field is located in the row area. A value of `0` or `false` indicates that this field is located in the column area. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataPosition (Default Data Field Position)	Specifies the position for the field representing multiple data field in the PivotTable, whether that field is located in the row area or column area. Missing attribute indicates this field is last, or innermost in the field list. 0 indicates this field is first, or outermost in the field list. 1 indicates this field is second in the field list. 2 indicates this field is third in the field list, and increasing values follow this pattern. If this value is higher than the number of fields in the field list, then this field is last, or innermost in the field list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
disableFieldList (Disable Field List)	Specifies a boolean value that indicates whether to disable the PivotTable field list. A value of `1` or `true` indicates the field list, or similar mechanism for selecting fields in the user interface, is disabled. A value of `0` or `false` indicates the field list is enabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
editData (Allow Edit Data)	Specifies a boolean value that indicates whether the user is allowed to edit the cells in the data area of the PivotTable. A value of `1` or `true` indicates the user can edit values in the data area. A value of `0` or `false` indicates the cells in the data area are not editable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
enableDrill (Enable Drill Down)	Specifies a boolean value that indicates whether the user is prevented from drilling down on a PivotItem or aggregate value. A value of `1` or `true` indicates the user can drill down on a pivot item or aggregate value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` indicates the user is prevented from drilling down pivot item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
enableFieldProperties (Enable Field Properties)	Specifies a boolean value that indicates whether the user is prevented from displaying PivotField properties. A value of `1` or `true` indicates the user can display pivot field properties. A value of `0` or `false` indicates the user cannot display pivot field properties. This attribute depends on how pivot field properties are exposed in the application user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
enableWizard (Enable PivotTable Editing Mechanism)	An implementation might choose to provide a mechanism for users to edit PivotTables. This attribute specifies a boolean value that indicates whether such a mechanism should be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
errorCaption (Error Caption)	Specifies the string to be displayed in cells that contain errors. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
fieldListSortAscending (Default Sort Order)	Specifies a boolean value that indicates whether fields in the PivotTable are sorted in non-default order in the field list. A value of `1` or `true` indicates fields for the PivotTable are sorted in the field list. The sort order from the data source is applied for range-based PivotTables. Alphabetical sorting is applied for external data PivotTables. A value of `0` or `false` indicates fields in the field list are not sorted. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fieldPrintTitles (Field Print Titles)	Specifies a boolean value that indicates whether the row and column titles from the PivotTable should be printed. A value of `1` or `true` indicates row and column titles should be printed. A value of `0` or `false` indicates row and column titles should not be printed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
grandTotalCaption (Grand Totals Caption)	Specifies the string to be displayed for grand totals. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
gridDropZones (Enable Drop Zones)	Specifies a boolean value that indicates whether the in-grid drop zones should be displayed at runtime, and whether classic layout is applied. A value of `1` or `true` indicates in-grid drop zones should be displayed and classic layout should be applied to the PivotTable. A value of `0` or `false` indicates in-grid drop zones should be disabled and classic layout should not be applied. [*Note:* Grid drop zones are optional runtime UI, determined by the application, that indicate to the user the locations of the page, row, column, and data fields in the PivotTable report. See layout discussion under pivotTableDefinition for the precise locations of these areas. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
immersive (Stop Immersive UI)	Specifies a boolean value that indicates whether PivotTable immersive experience user interface should be turned off. A value of `1` or `true` indicates the PivotTable immersive experience should be turned off for this PivotTable. A value of `0` or `false` indicates the immersive experience should be left on. This attribute depends on whether the application implements an immersive experience in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indentation for Compact Axis)	Specifies the indentation increment for compact axis and can be used to set the Report Layout to Compact Form. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
itemPrintTitles (Item Print Titles)	Specifies a boolean value that indicates whether PivotItem names should be repeated at the top of each printed page. A value of `1` or `true` indicates pivot items names should be repeated at the top of each

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	page. A value of `0` or `false` indicates should not be repeated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
mdxSubqueries (MDX Subqueries Supported)	Specifies a boolean value that indicates whether MDX sub-queries are supported by OLAP data provider for this PivotTable. A value of `1` or `true` indicates MDX sub-queries are supported by the OLAP data provider. A value of `0` or `false` indicates MDX sub-queries are not supported. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
mergeItem (Merge Titles)	Specifies a boolean value that indicates whether row or column titles that span multiple cells should be merged into a single cell. A value of `1` or `true` indicates that titles that span multiple cells are merged into a single cell. A value of `0` or `false` indicates titles are not merged. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minRefreshableVersion (Minimum Refreshable Version)	Specifies the minimum version of the application required to update this PivotTable view. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
missingCaption (Caption for Missing Values)	Specifies the string to be displayed in cells with no value The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
multipleFieldFilters (Multiple Field Filters)	Specifies a boolean value that indicates whether the fields of a PivotTable can have multiple filters set on them. A value of `1` or `true` indicates the fields of a PivotTable can have multiple filters. A value of `0` or `false` indicates the fields of a PivotTable can only have a simple filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Name)	Specifies the PivotTable name.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
outline (Outline New Fields)	Specifies a boolean value that indicates whether new fields should have their `outline` flag set to true. A value of `1` or `true` indicates new fields are created with `outline` equal to true. A value of `0` or `false` indicates new fields are created with `outline` equal to false. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
outlineData (Outline Data Fields)	Specifies a boolean value that indicates whether data fields in the PivotTable should be displayed in outline form. A value of `1` or `true` indicates data fields will display in outline form. A value of `0` or `false` indicates data fields will not display in outline form. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pageOverThenDown (Page Over Then Down)	Specifies a boolean value that indicates how the page fields are laid out when there are multiple PivotFields in the page area. A value of `1` or `true` indicates the fields will display "Over, then down" A value of `0` or `false` indicates the fields will display "down, then Over" The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
pageStyle (Page Header Style Name)	Specifies the name of the style to apply to each of the field item headers in the page area of the PivotTable. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
pageWrap (Page Wrap)	Specifies the number of page fields to display before starting another row or column. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
pivotTableStyle (Table Style Name)	Specifies the name of the style to apply to the main table area of the PivotTable. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
preserveFormattin	Specifies a boolean value that indicates whether the formatting applied by the user to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
g (Preserve Formatting)	the PivotTable cells is discarded on refresh. A value of `1` or `true` indicates the formatting applied by the end user is discarded on refresh. A value of `0` or `false` indicates the end-user formatting is retained on refresh. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
printDrill (Print Drill Indicators)	Specifies a boolean value that indicates whether drill indicators expand collapse buttons should be printed. A value of `1` or `true` indicates that these buttons should be printed. A value of `0` or `false` indicates that these buttons should not be printed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
published (Data Fields Published)	Specifies a boolean value that indicates whether data fields in the PivotTable are published and available for viewing in a server rendering environment. A value of `1` or `true` indicates that the data fields in the PivotTable are published and shall be available for viewing in a server rendering environment. A value of `0` or `false` indicates that the data fields in the PivotTable are not published and shall not be available for viewing in a server rendering environment. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowGrandTotals (Row Grand Totals)	Specifies a boolean value that indicates whether grand totals should be displayed for the PivotTable rows. The default value for this attribute is true. A value of `1` or `true` indicates grand totals are displayed for the PivotTable rows. A value of `0` or `false` indicates grand totals will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowHeaderCaption (Row Header Caption)	Specifies the string to be displayed in row header in `compact` mode. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
showCalcMbrs (Show Calculated	Specifies a boolean value that indicates whether calculated members should be shown in the PivotTable view. This attribute applies to PivotTables from OLAP-sources only.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Members)	A value of 1 or true indicates that calculated members should be shown. A value of 0 or false indicates calculated members should not be shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showDataDropDown (Show Drop Down)	Specifies a boolean value that indicates whether the drop-down lists for the fields in the PivotTable should be hidden. This attribute depends on whether the application implements drop down lists or similar mechanism in the user interface. A value of 1 or true indicates drop down lists are displayed for fields. A value of 0 or false indicates drop down lists will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showDataTips (Show ToolTips on Data)	Specifies a boolean value that indicates whether tooltips should be displayed for PivotTable data cells. A value of 1 or true indicates tooltips are displayed. A value of 0 or false indicates tooltips will not be displayed. This attribute depends on whether the application employs tooltips or similar mechanism in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showDrill (Show Expand Collapse)	Specifies a boolean value that indicates whether drill indicators should be hidden. A value of 1 or true indicates drill indicators are displayed. A value of 0 or false indicates drill indicators will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showDropZones (Show Drop Zones)	Specifies a boolean value that indicates whether the PivotTable should display large drop zones when there are no fields in the data region. A value of 1 or true indicates a large drop zone is displayed. A value of 0 or false indicates a large drop zone will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
showEmptyCol (Show Empty Column)	Specifies a boolean value that indicates whether to include empty columns in the table. A value of `1` or `true` indicates empty columns are included in the PivotTable. A value of `0` or `false` indicates empty columns are excluded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showEmptyRow (Show Empty Row)	Specifies a boolean value that indicates whether to include empty rows in the table. A value of `1` or `true` indicates empty rows are included in the PivotTable. A value of `0` or `false` indicates empty rows are excluded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showError (Show Error)	Specifies a boolean value that indicates whether to show error messages in cells. A value of `1` or `true` indicates error messages are shown in cells. A value of `0` or `false` indicates error messages are shown through another mechanism the application provides in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showHeaders (Show Field Headers)	Specifies a boolean value that indicates whether to suppress display of pivot field headers. A value of `1` or `true` indicates field headers are shown in the PivotTable. A value of `0` or `false` indicates field headers are excluded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showItems (Show Item Names)	Specifies a boolean value that indicates whether to display item names when adding a field onto a PivotTable that has no data fields. A value of `1` or `true` indicates item names are displayed. A value of `0` or `false` indicates item names will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMemberProp	Specifies a boolean value that indicates whether member property information should be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
ertyTips (Show Member Property ToolTips)	omitted from PivotTable tooltips. A value of `1` or `true` indicates member property information is included. A value of `0` or `false` indicates member property information is excluded. This attribute depends on whether the application employs tooltips or similar mechanism in the user interface. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMissing (Show Missing)	Specifies a boolean value that indicates whether to show a message in cells with no value. A value of `1` or `true` indicates to show a message string in cells without values. A value of `0` or `false` indicates no message string will shown in cells without values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMultipleLabel (Show Multiple Labels)	Specifies a boolean value that indicates whether a page field with multiple selected items should display "(multiple items)" instead of "All". This attribute applies only to non-OLAP PivotTables. The messages displayed depend on the application implementation. A value of `1` or `true` indicates a different message string is displayed for a page field with multiple items. A value of `0` or `false` indicates the same message string is displayed for all page fields. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
subtotalHiddenItems (Subtotal Hidden Items)	Specifies a boolean value that indicates whether data for hidden pivotItems for PivotFields in the data area should be included in subtotals. A value of `1` or `true` indicates that data for hidden pivot items in the data area is included in subtotals. A value of `0` or `false` indicates hidden pivot items will not be included in subtotals. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tag (PivotTable Custom String)	Specifies a user-defined string that is associated with this PivotTable. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
updatedVersion (PivotTable Last Updated Version)	Specifies the version of the application that last updated the PivotTable view. This attribute is application-dependent. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
useAutoFormatting (Auto Formatting)	Specifies a boolean value that indicates whether legacy auto formatting has been applied to the PivotTable view. A value of `1` or `true` indicates that legacy auto formatting has been applied to the PivotTable. A value of `0` or `false` indicates that legacy auto formatting has not been applied to the PivotTable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vacatedStyle (Vacated Style)	Specifies the name of the style to apply to the cells left blank when a PivotTable shrinks during a refresh operation The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
visualTotals (Total Visual Data)	Specifies a boolean value that indicates whether totals should be based on visible data only. This attribute applies to OLAP PivotTables only. A value of `1` or `true` indicates subtotals are computed on visible data only. A value of `0` or `false` indicates subtotals are computed on all data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_pivotTableDefinition) is located in §A.2. *end note*]

18.10.1.74 pivotTableStyleInfo (PivotTable Style)

Represent information on style applied to the PivotTable.

[*Example*:

```
<sh:pivotTableStyleInfo name="PivotStyleLight16" showRowHeaders="1"
  showColHeaders="1" showRowStripes="0" showColStripes="0" showLastColumn="1"/>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
name (Table Style Name)	Specifies the name of the table style to use with this table. The possible values for this attribute are defined by the W3C XML Schema string datatype.
showColHeaders (Show Table Style Column Header Formatting)	Specifies a boolean value that indicates whether to show column headers for the table. A value of `1` or `true` indicates column headers are shown. A value of `0` or `false` indicates column headers are omitted. 'True' if table style column header formatting should be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showColStripes (Show Column Stripes)	Specifies a boolean value that indicates whether to show column stripe formatting for the table. A value of `1` or `true` indicates column stripe formatting is shown. A value of `0` or `false` indicates no column formatting is shown. True if table style column stripe formatting should be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showLastColumn (Show Last Column)	Specifies a boolean value that indicates whether to show the last column. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRowHeaders (Show Row Header Formatting)	Specifies a boolean value that indicates whether to show row headers for the table. A value of `1` or `true` indicates table style formatting is displayed. A value of `0` or `false` indicates table style formatting will not be displayed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showRowStripes (Show Row Stripes)	Specifies a boolean value that indicates whether to show row stripe formatting for the table. A value of `1` or `true` indicates row stripe formatting is displayed. A value of `0` or `false` indicates no row formatting is shown.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotTableStyle) is located in §A.2. *end note*]

18.10.1.75 query (Query)

Represents an OLAP sheet data cached query.

Attributes	Description
mdx (MDX Query String)	Specifies the Multidimensional Expressions (MDX) query string. [*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn2.microsoft.com/en-us/library/ms145595.aspx *end note*] See the MDX Language Reference for more information: The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Query) is located in §A.2. *end note*]

18.10.1.76 queryCache (OLAP Query Cache)

Represents the cache of OLAP sheet data queries.

Attributes	Description
count (Cached Query Count)	Specifies the number of cached queries in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryCache) is located in §A.2. *end note*]

18.10.1.77 r (PivotCache Record)

Represents a single record of data in the PivotCache.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<r>
  <s v="3550"/>
  <s v="Road-150 Red, 62"/>
  <s v="This bike is ridden by race winners. Developed with the Adventure Works
    Cycles professional race team, it has a extremely light heat-treated
    aluminum frame, and steering that allows precision control."/>
  <s v="No Discount"/>
  <x v="0"/>
  <s v="Australian Dollar"/>
  <n v="1"/>
  <n v="3578.27"/>
  <n v="0"/>
  <n v="2171.2941999999998"/>
  <n v="3578.27"/>
  <n v="89.456800000000001"/>
</r>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Record) is located in §A.2. *end note*]

18.10.1.78 rangePr (Range Grouping Properties)

Represents the collection of range grouping properties.

[*Example*:

```
<rangePr groupBy="months" startDate="2002-01-01T00:00:00"
  endDate="2006-05-06T00:00:00"/>
```

end example]

Attributes	Description
autoEnd (Source Data Ending Range)	Specifies a boolean value that indicates whether the application uses the source data to set the ending range value. A value of `1` or `true` indicates the ending range value is set from the source data. A value of `0` or `false` indicates ending range values are set by the value specified in endDate or endNum. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoStart (Source Data Set Beginning	Specifies a boolean value that indicates whether we use source data to set the beginning range value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Range)	A value of 1 or true indicates the beginning range value is set from the source data. A value of 0 or false indicates the beginning range value is set from the value specified in startDate or startNum. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
endDate (Date Grouping End Value)	Specifies the ending value for date grouping if autoEnd is false. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
endNum (Numeric Grouping End Value)	Specifies the ending value for numeric grouping if autoEnd is false. The possible values for this attribute are defined by the W3C XML Schema double datatype.
groupBy (Group By)	Specifies the grouping. The possible values for this attribute are defined by the ST_GroupBy simple type (§18.18.38).
groupInterval (Grouping Interval)	Specifies the grouping interval for numeric range grouping. Specifies the number of days to group by in date range grouping. The possible values for this attribute are defined by the W3C XML Schema double datatype.
startDate (Date Grouping Start Value)	Specifies the starting value for date grouping if autoStart is false. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
startNum (Numeric Grouping Start Value)	Specifies the starting value for numeric grouping if autoStart is false. The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_RangePr) is located in §A.2. *end note*]

18.10.1.79 rangeSet (Range Set)

Represents a single range in the rangeSets collection. element is intended to facilitate creating a PivotTable report by consolidating SpreadsheetML ranges that have similar categories of data to be summarized. The simplest layout for the data source is for each rangeSets of data to be in list-like format, with column labels in the first row, row labels in the first column, the rest of the rows having similar items in the same row and

 ©ISO/IEC 2016 – All rights reserved

column, and no blank rows or columns within the range. A particular rangeSet can consist of a built-in named range that is provided by the application, a user defined named range, a range reference, or a reference to an external workbook.

When multiple ranges are consolidated using this functionality, up to 4 custom report filters (also known as page fields) can be created to help filter the PivotTable report, by specifically enabling one or more of the individual ranges to be selected in the report filter. For each custom page field created, a custom label can be specified and assigned to each range participating in the consolidation range, so that the PivotTable can be filtered by one or more of the ranges being summarized.

[*Example*: Consider a workbook with 6 worksheets. On Sheet1 we have:

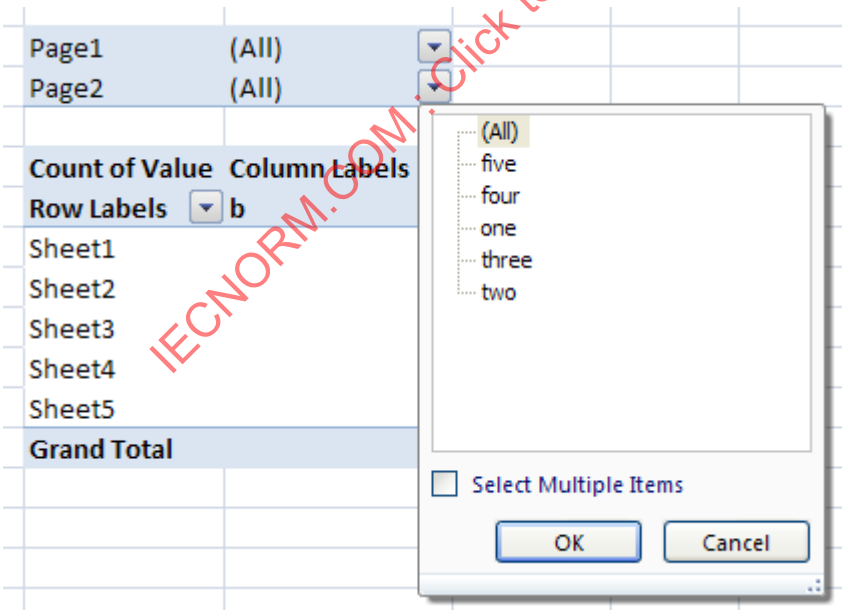
a	b
Sheet1	Sheet1
Sheet1	Sheet1

On Sheet2 we have:

a	b
Sheet2	Sheet2
Sheet2	Sheet2

… and so on up through Sheet5.

On Sheet6, we have the consolidated ranges being summarized by a PivotTable, and two page filters exist for the PivotTable.



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Notice that for the second page filter, the items have been assigned a custom label, "one", "two", …, "five", for each of Sheet1, Sheet2, …, Sheet5 data sources, respectively. Similarly, the items have been assigned a custom label, "1", "2", …, "5" for each of Sheet1, Sheet2, …, Shet5 data sources, respectively.

The XML representing these custom page filters must be like the following:

```
<cacheSource type="consolidation">
  <consolidation autoPage="0">
    <pages count="2">
      <page count="5">
        <pageItem name="1"/>
        <pageItem name="2"/>
        <pageItem name="3"/>
        <pageItem name="4"/>
        <pageItem name="5"/>
      </page>
      <page count="5">
        <pageItem name="one"/>
        <pageItem name="two"/>
        <pageItem name="three"/>
        <pageItem name="four"/>
        <pageItem name="five"/>
      </page>
    </pages>
    <rangeSets count="5">
      <rangeSet i1="0" i2="0" ref="A1:B3" sheet="Sheet1"/>
      <rangeSet i1="1" i2="1" ref="A1:B3" sheet="Sheet2"/>
      <rangeSet i1="2" i2="2" ref="A1:B3" sheet="Sheet3"/>
      <rangeSet i1="3" i2="3" ref="A1:B3" sheet="Sheet4"/>
      <rangeSet i1="4" i2="4" ref="A1:B3" sheet="Sheet5"/>
    </rangeSets>
  </consolidation>
</cacheSource>
```

end example]

[*Note*: Attributes i1, i2, i3, and i4 correspond to custom page fields created in the user interface. Spreadsheet ML only supports 4 custom page fields. *end note*]

Attributes	Description
i1 (Field Item Index Page 1)	Specifies the index of a page field item in page filter one. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
i2 (Field Item Index Page 2)	Specifies the index of a page field item in page filter two. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
i3 (Field Item index Page 3)	Specifies the index of a page field item in page filter three. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
i4 (Field Item Index Page 4)	Specifies the index of a page field item in page filter four. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the unique identifier of the Workbook part where the range set is stored. See Workbook (§18.2) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Named Range)	Specifies the named range. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ref (Reference)	Specifies the cell range. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sheet (Sheet Name)	Specifies the sheet name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RangeSet) is located in §A.2. *end note*]

18.10.1.80 rangeSets (Range Sets)

Represents the collection of reference-page items pairs.

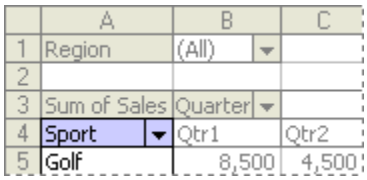
Attributes	Description
count (Reference and Page Item Count)	Specifies the number of reference and page items. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_RangeSets) is located in §A.2. *end note*]

18.10.1.81 rowFields (Row Fields)

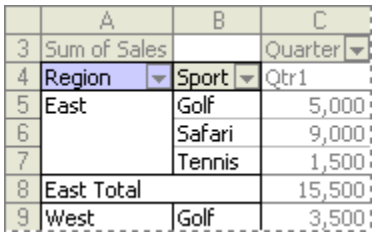
Represents the collection of row fields for the PivotTable.



	A	B	C
1	Region	(All)	
2			
3	Sum of Sales	Quarter	
4	Sport	Qtr1	Qtr2
5	Golf	8,500	4,500

In the image above, the blue field is a row field. A PivotTable report that has more than one row field has one inner row field (Sport, in the example below), the one closest to the data area. Any other row fields are outer row fields (Region, in the example below). Items in the outermost row field are displayed only once, but items in the rest of the row fields are repeated as needed.

[*Example*:



	A	B	C
3	Sum of Sales		Quarter
4	Region	Sport	Qtr1
5	East	Golf	5,000
6		Safari	9,000
7		Tennis	1,500
8	East Total		15,500
9	West	Golf	3,500

In the image above, Region is an outer row field. Sport is an inner row field.

```
<rowFields count="2">
  <field x="7"/>
  <field x="8"/>
</rowFields>
```

end example]

Attributes	Description
count (Repeated Items Count)	Specifies the number of repeated items in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RowFields) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.82 rowHierarchiesUsage (Row OLAP Hierarchy References)

Represents the collection of references to OLAP hierarchies on the row axis of a PivotTable.

[*Example*:

```
<sh:rowHierarchiesUsage count="1">
  <sh:rowHierarchyUsage hierarchyUsage="9"/>
</sh:rowHierarchiesUsage>
```

end example]

Attributes	Description
count (Item Count)	Specifies the number of items in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RowHierarchiesUsage) is located in §A.2. *end note*]

18.10.1.83 rowHierarchyUsage (Row OLAP Hierarchies)

Represents a references to an OLAP Hierarchy on the row axis of a PivotTable.

[*Example*:

```
<sh:rowHierarchyUsage hierarchyUsage="9"/>
```

end example]

Attributes	Description
hierarchyUsage (Hierarchy Usage)	Specifies the reference to an OLAP hierarchy in a PivotTable. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HierarchyUsage) is located in §A.2. *end note*]

18.10.1.84 rowItems (Row Items)

Represents the collection of items in row axis of the PivotTable.

[*Example*: In the SpreadsheetML example below, the item values are found in cells B10:B13. For example "Bikes" is in B10, and corresponds to the first `<i>` element below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<rowItems count="4">
  <i>
    <x/>
  </i>
  <i r="1">
    <x/>
  </i>
  <i r="1">
    <x v="1"/>
  </i>
  <i t="grand">
    <x/>
  </i>
</rowItems>
```

Looking at the layout of the PivotTable in this example, "Bikes" is the first (and only) item value in the first row, in cell B10. In the XML defining the PivotTable row item values, the first <i> element corresponds to the first row. There is a single index element <x>. The first (and only) <x> element corresponds to the first field on the row axis, namely "Product Category", and an index value of "0" indicates that the 0th item in the items collection for that pivotField definition is how to obtain the item value. Note that "Bikes" isn't explicitly listed as a value here, but instead the 0th item is an index to this field's shared items collection in the pivotCacheDefinition part.

For the second row there are two item values, one item value (Bikes) from the first field in that row (Product Category) and one item value (Mountain Bikes) from the second field in that row (Product Subcategory). In the PivotTable, the first item value "Bikes" is hidden from view. In the XML for this example, the second <i> element expresses both item values for this row. The first item value "Bikes" is expressed implicitly, because the value of @r on the second <i> element is '1', indicating that the first item value from the previous row is reused again as the first item value for the current row. The second item value is expressed explicitly via the <x> element under the second <i> element. The index of '0' indicates that the 0th item in the <pivotField> element for that field is how to obtain the item value. Note again that the 0th item is itself an index into this field's shared items collection in the pivotCacheDefinition part.

The item values for the third row can be discovered in a similar way. *end example*]

Attributes	Description
count (Items in a Row Count)	Specifies the number of items in the row axis of the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_rowItems) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.10.1.85 s (Character Value)

Represents a character value in a PivotTable.

[*Example*:

```
<sharedItems count="2">
  <s v="7527 Brook Way"/>
  <s v="3310 Harvey Way"/>
</sharedItems>
```

end example]

Attributes	Description
b (Bold)	Specifies a boolean value that indicates whether this value contains bold formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of `1` or `true` indicates this value contains bold formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bc (Background Color)	Specifies the background color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space. The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
c (Item Caption)	Specifies the caption for the this item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cp (Member Property Count)	Specifies the number of member property values. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
f (Calculated Item)	Specifies a boolean value that indicates whether this is a calculated item value. A value of `1` or `true` indicates this item is a calculated value. A value of `0` or `false` indicates this item is not a calculated value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fc (Foreground Color)	Specifies the foreground color for this value that was provided by the OLAP server. This attribute applies to OLAP-based PivotTables only. The color is specified as a HEX value in RGB space.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_UnsignedIntHex simple type (§18.18.86).
i (Italic)	Specifies a boolean value that indicates whether the value contains italic formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of 1 or true indicates this value contains italic formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
in (Format Index)	Specifies the index to the OLAP serverformat element where the format string for this entry is stored. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
st (Strikethrough)	Specifies a boolean value that indicates whether the value contains strikethrough formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of 1 or true indicates this value contains strikethrough formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
u (Unused Item)	Specifies a boolean value that indicates whether this is an unused item. The application marks an item as unused when an item is deleted from the data source. The item and associated metadata are retained in the cache until the threshold for unused items specified in missingItemsLimit is reached. A value of 1 or true indicates this item is unused. A value of 0 or false indicates this item is used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
un (Underline)	Specifies a boolean value that indicates whether the value contains underline formatting on the OLAP server. This attribute applies to OLAP-based PivotTables only. A value of 1 or true indicates this value contains underline formatting on the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
v (Value)	Specifies the value of the item. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.2. *end note*]

18.10.1.86 serverFormat (Server Format)

Represents the numeric format specified by the OLAP server for a tuple.

Attributes	Description
culture (Culture)	Specifies a language used to determine the currency symbol to display for currency values. [*Example*: if the culture is "en-us", the values in the application will format the values with a dollar sign. If the culture is "fr-fr" the application will format the values with a euro sign. *end example*] This value conforms to the language tagging conventions of RFC 3066 and later. The pattern <language>-<REGION> is used, e.g., "en-us" or "fr-fr". The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
format (Format)	Specifies the format string to use for all other numeric values. This string is supplied by the OLAP server. Therefore, the syntax for reading the format string depends on the server implementation. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ServerFormat) is located in §A.2. *end note*]

18.10.1.87 serverFormats (Server Formats)

Represents the collection of numeric and currency formats specified by the OLAP server for a tuple

Attributes	Description
count (Format Count)	Specifies the number of formats in the collection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ServerFormats) is located in §A.2. *end note*]

18.10.1.88 set (OLAP Set)

Represents an OLAP sheet data set or tuple set. The set is defined by a Multidimensional Expressions (MDX) query that specifies criteria for the dimension members that belong to the set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: the following MDX expression defines the set for the 10 salespersons with the lowest sales:

BottomCount([Salesperson].[Salesperson Name].Members,10,[Measures].[Sales])

end example]

The MDX expression is specified in the setDefinition attribute.

Attributes	Description
count (Number of Tuples)	Specifies the number of tuples in the set. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
maxRank (Maximum Rank Requested)	Specifies the largest rank entry the user has requested. The possible values for this attribute are defined by the W3C XML Schema int datatype.
queryFailed (Query Failed)	Specifies a boolean value that indicates whether querying on this set failed. A value of `1` or `true` indicates a query against this set failed. A value of `0` or `false` indicates a query against this set succeeded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
setDefinition (MDX Set Definition)	Specifies the Multidimensional Expressions (MDX) set definition. [*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn2.microsoft.com/en-us/library/ms145595.aspx *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sortType (Set Sort Order)	Specifies the sort order of the set. The possible values for this attribute are defined by the ST_SortType simple type (§18.18.74).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Set) is located in §A.2. *end note*]

18.10.1.89 sets (Sets)

Represents the collection of OLAP sheet data entries or tuple sets.

Attributes	Description
count (Tuple Set	Specifies the number of tuple sets.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Count)	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sets) is located in §A.2. *end note*]

18.10.1.90 sharedItems (Shared Items)

Represents the collection of unique items for a field in the PivotCacheDefinition. The sharedItems complex type stores data type and formatting information about the data in a field. Items in the PivotCacheDefinition can be shared in order to reduce the redundancy of those values that are referenced in multiple places across all the PivotTable parts. [*Example*: A value might be part of a filter, it might appear on a row or column axis, and will appear in the pivotCacheRecords definition as well. However, because of the performance cost of creating the optimized shared items, items are only shared if they are actually in use in the PivotTable. Therefore, depending on user actions on the PivotTable layout, the pivotCacheDefinition and underlying PivotCacheRecords part can be updated. *end example*]

If there are no shared items, then field values are stored directly in the pivotCacheRecords part.

[*Example*:

```
<sharedItems count="1">
  <s v="[Customer].[Customer Geography].[Country].&amp;[United States]"
    c="United States"/>
</sharedItems>
```

end example]

The following attributes are not required or used if there are no items in sharedItems.

- containsBlank
- containsSemiMixedTypes
- containsMixedTypes
- longText

The following attributes are not used unless there is more than one item in sharedItems or the one and only item is not a blank item. If the first item is a blank item the data type the field cannot be verified.

- containsNumber
- containsDates
- containsString
- containsInteger

The following attributes can be omitted without loss of functionality.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- containsNonDate
- count

The following attributes are not required and can be omitted. However, refreshing the PivotTable could produce different groupings than before.

- maxDate
- minDate
- maxValue
- minValue

Applications should ensure that “date” attributes are not mixed with “value” attributes.

Attributes	Description
containsBlank (Contains Blank)	Specifies a boolean value that indicates whether this field contains a blank value. A value of `1` or `true` indicates this field contains one or more blank values. A value of `0` or `false` indicates this field does not contain blank values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsDate (Contains Date)	Specifies a boolean value that indicates that the field contains at least one date. A value of `1` or `true` indicates the field contains at least one date value. A value of `0` or `false` indicates the field does not contain any date values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsInteger (Contains Integer)	Specifies a boolean value that indicates whether this field contains integer values. A value of `1` or `true` indicates this field contains integer values. A value of `0` or `false` indicates non-integer or mixed values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsMixedTypes (Contains Mixed Data Types)	Specifies a boolean value that indicates whether this field contains more than one data type. A value of `1` or `true` indicates this field contains more than one data type. A value of `0` or `false` indicates contains only one data type. The field can still contain blank values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsNonDate (Contains Non Date)	Specifies a boolean value that indicates that the field contains at least one value that is not a date. A value of `1` or `true` indicates the field contains at least one non-date values. A value of `0` or `false` indicates this field contains no date fields. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsNumber (Contains Numbers)	Specifies a boolean value that indicates whether this field contains numeric values. A value of `1` or `true` indicates this field contains at least one numeric value. A value of `0` or `false` indicates this field contains no numeric values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsSemiMixedTypes (Contains Semi Mixed Data Types)	Specifies a boolean value that indicates that this field contains text values. The field can also contain a mix of other data types and blank values. A value of `1` or `true` indicates at least one text value, and can also contain a mix of other data types and blank values. A value of `0` or `false` indicates the field does not have a mix of text and other values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
containsString (Contains String)	Specifies a boolean value that indicates whether this field contains a text value. A value of `1` or `true` indicates this field contains at least one text value. A value of `0` or `false` indicates this field does not contain any text values. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
count (Shared Items Count)	Specifies the number of shared items to load for this field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
longText (Long Text)	Specifies a boolean value that indicates whether this field contains a long text value. A string is considered long if it is over 255 Unicode scalar values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `1` or `true` indicates the value contains more than 255 Unicode scalar valuesof text. A value of `0` or `false` indicates the value contains less than 255 Unicode scalar values. [*Note*: This is used as many legacy spreadsheet application support a limit of 255 characters for text values. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
maxDate (Maximum Date Time Value)	Specifies the maximum date/time value found in a date field. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
maxValue (Maximum Numeric Value)	Specifies the maximum numeric value found in a numeric field. The possible values for this attribute are defined by the W3C XML Schema double datatype.
minDate (Minimum Date Time)	Specifies the minimum date/time value found in a date field. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
minValue (Minimum Numeric Value)	Specifies the minimum numeric value found in a numeric field. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SharedItems) is located in §A.2. *end note*]

18.10.1.91 sortByTuple (Sort By Tuple)

Represents the sort applied to a tuple.

Attributes	Description
c (Member Name Count)	Specifies the number of member names. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tuples) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.10.1.92 tpl (Tuple)

Represents an OLAP sheet data entry member.

Attributes	Description
fld (Field Index)	Specifies the index of the field to which the member belongs. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
hier (Hierarchy Index)	Specified the index of the hierarchy to which the member belongs. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
item (Item Index)	Specifies the index of the item in the field that represents this item. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tuple) is located in §A.2. *end note*]

18.10.1.93 tpls (Tuples)

Represents members for the OLAP sheet data entry, also known as a tuple.

Attributes	Description
c (Member Name Count)	Specifies the number of member names. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tuples) is located in §A.2. *end note*]

18.10.1.94 tupleCache (Tuple Cache)

Represents the cache of OLAP sheet data members, or tuples.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TupleCache) is located in §A.2. *end note*]

18.10.1.95 worksheetSource (Worksheet PivotCache Source)

Represents the location of the source of the data that is stored in the cache.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<cacheSource type="worksheet">
  <worksheetSource name="Table1" r:id="rId2"/>
</cacheSource>
```

end example]

Attributes	Description
id (Relationship Id) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identifier to the Sheet part whose data is stored in the cache. See the Sheet section (§18.2) for more information. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Named Range)	Specifies the named range that is the source of the data. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ref (Reference)	Specifies the reference that defines a cell range that is the source of the data. This attribute depends on how the application implements cell references. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sheet (Sheet Name)	Specifies the name of the sheet that is the source for the cached data. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WorksheetSource) is located in §A.2. *end note*]

18.10.1.96 x (Member Property Index)

Represents an array of indexes to cached member property values.

Attributes	Description
v (Shared Items Index)	Specifies the index into the shared items table in the PivotCache that identifies this item. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_X) is located in §A.2. *end note*]

18.10.1.97 x (Shared Items Index)

This element represents an array of indexes to cached shared item values

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
v (Shared Items Index)	Specifies the index into the shared items table in the PivotCache that identifies this item. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Index) is located in §A.2. *end note*]

18.10.2 Shared Pivot Table Data

This section defines the part where shared PivotTable data is stored.

18.10.2.1 reference (Reference)

Represents a set of selected fields and selected items within those fields.

[*Example*:

```
<sh:reference field="4294967294" count="1" selected="0">
  <sh:x v="0"/>
</sh:reference>
```

end example]

Attributes	Description
avgSubtotal (Include Average Filter)	Specifies a boolean value that indicates whether the 'average' aggregate function is included in the filter. A value of `1` or `true` indicates the average aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
byPosition (Positional Reference)	Specifies a boolean value that indicates whether the item is referred to by position rather than item index. A value of `1` or `true` indicates the item is referred to by position. A value of `0` or `false` indicates the item is referred to by index. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
count (Item Index Count)	Specifies the number of item indexes in the collection of indexes (x tags). The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
countASubtotal (Include CountA Filter)	Specifies a boolean value that indicates whether the 'countA' subtotal is included in the filter. A value of `1` or `true` indicates the count aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
countSubtotal (Include Count Subtotal)	Specifies a boolean value that indicates whether the count aggregate function is included in the filter. A value of `1` or `true` indicates the count aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
defaultSubtotal (Include Default Filter)	Specifies a boolean value that indicates whether the default subtotal is included in the filter. A value of `1` or `true` indicates the default subtotal is included in the filter. The default is to display the total or the grand total. A value of `0` or `false` indicates another subtotal or aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
field (Field Index)	Specifies the index of the field to which this filter refers. A value of -2 indicates the 'data' field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
maxSubtotal (Include Maximum Filter)	Specifies a boolean value that indicates whether the 'maximum' aggregate function is included in the filter. A value of `1` or `true` indicates the maximum aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
minSubtotal (Include Minimum Filter)	Specifies a boolean value that indicates whether the 'minimum' aggregate function is included in the filter. A value of `1` or `true` indicates the minimum aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
productSubtotal (Include Product Filter)	Specifies a boolean value that indicates whether the 'product' aggregate function is included in the filter. A value of `1` or `true` indicates the product aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
relative (Relative Reference)	Specifies a boolean value that indicates whether the item is referred to by a relative reference rather than an absolute reference. This attribute is used if `posRef` is set to true. A value of `1` or `true` indicates the item is referred to by a relative reference. A value of `0` or `false` indicates the item is referred to by an absolute reference. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
selected (Selected)	Specifies a boolean value that indicates whether this field has selection. This attribute is used when the PivotTable is in Outline view. It is also used when both header and data cells have selection. A value of `1` or `true` indicates the field has selection. A value of `0` or `false` indicates the field does not have selection. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
stdDevPSubtotal (Include StdDevP Filter)	Specifies a boolean value that indicates whether the population standard deviation aggregate function is included in the filter. A value of `1` or `true` indicates the population standard deviation aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
stdDevSubtotal (Include StdDev Filter)	Specifies a boolean value that indicates whether the standard deviation aggregate function is included in the filter. A value of `1` or `true` indicates the standard deviation aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sumSubtotal (Include Sum Filter)	Specifies a boolean value that indicates whether the sum aggregate function is included in the filter. A value of `1` or `true` indicates the sum aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
varPSubtotal (Include VarP Filter)	Specifies a boolean value that indicates whether the population variance aggregate function is included in the filter. A value of `1` or `true` indicates the population variance aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
varSubtotal (Include Var Filter)	Specifies a boolean value that indicates whether the variance aggregate function is included in the filter. A value of `1` or `true` indicates the variance aggregation function is included in the filter. A value of `0` or `false` indicates another aggregation function is included in the filter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotAreaReference) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.10.2.2 references (References)

Represents the set of selected fields and the selected items within those fields.

[*Example*:

```
<sh:references count="5">
  <sh:reference field="4294967294" count="1" selected="0">
    <sh:x v="0"/>
  </sh:reference>
  <sh:reference field="2" count="1" selected="0">
    <sh:x v="0"/>
  </sh:reference>
  <sh:reference field="14" count="1" selected="0">
    <sh:x v="0"/>
  </sh:reference>
  <sh:reference field="15" count="2" selected="0">
    <sh:x v="2"/>
    <sh:x v="3"/>
  </sh:reference>
</sh:references>
```

end example]

Attributes	Description
count (Pivot Filter Count)	Specifies the number of filtered records available in the PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotAreaReferences) is located in §A.2. *end note*]

18.11 Shared Workbook Data

The Shared Workbooks architecture enables a spreadsheet application to record revisions made to a workbook (e.g., track changes), and is designed to enable either single, or multiple users editing the same workbook at the same time. Therefore, the application needs to support the ability to read changes made by another user and update its own state of the same workbook with those changes, even when those changes are made concurrently with other changes made by other users. Inevitably there are conflicts, and therefore merge conflict resolution should be supported the runtime application. The file format only contains enough information so that the spreadsheet application can deal with conflicts, and can undo/redo changes from the change history at run time.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.11.1 Shared Workbook Data

Within a shared workbook, the changes made to the spreadsheet at runtime are persisted as sets of different revisions collectively forming a revision history. These are persisted to the file on disk during a save event, and are saved in different xml parts known as revision logs. There is a headers table xml part that summarizes when changes were made, who made them, and it lists the relationship from each header to the individual revision log that records the specific changes.

[*Example*: This example shows the header and revision log for two simple events: adding text to a cell, and inserting a new sheet.

First, take a look at the header table, and revision log:

```
<headers xmlns="…" xmlns:r="…" guid="{A84A6777-8908-4CB9-9EB6-625CEFF419D3}">
  <header guid="{A84A6777-8908-4CB9-9EB6-625CEFF419D3}"
    dateTime="2006-07-14T13:42:54" maxSheetId="4" userName="UserName"
    r:id="rId1">
    <sheetIdMap count="3">
      <sheetId val="1"/>
      <sheetId val="2"/>
      <sheetId val="3"/>
    </sheetIdMap>
  </header>
</headers>
```

And the revision log is essentially empty:

```
<revisions xmlns="…" xmlns:r="…"/>
```

Now, after inserting the text "foo" into cell A1, and saving, the header looks like this:

```
<headers xmlns="…" xmlns:r="…" guid="{CFEA9B63-728B-4274-A346-0440E1573AB4}"
  diskRevisions="1" revisionId="1" version="2">
  <header guid="{A84A6777-8908-4CB9-9EB6-625CEFF419D3}"
    dateTime="2006-07-14T13:42:54" maxSheetId="4" userName="UserName"
    r:id="rId1">
    <sheetIdMap count="3">
      <sheetId val="1"/>
      <sheetId val="2"/>
      <sheetId val="3"/>
    </sheetIdMap>
  </header>
  <header guid="{CFEA9B63-728B-4274-A346-0440E1573AB4}"
    dateTime="2006-07-14T13:44:40" maxSheetId="4" userName="UserName"
    r:id="rId2" minRId="1">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <sheetIdMap count="3">
        <sheetId val="1"/>
        <sheetId val="2"/>
        <sheetId val="3"/>
      </sheetIdMap>
    </header>
  </headers>
```

A new header entry is added, with a GUID and a revision ID (rId2) that specifies which log to look into to see the details about the revision.

The old log is saved, and the newly created log (corresponding to rId2) now looks like this:

```
<revisions xmlns="…" xmlns:r="…">
  <rcc rId="1" sId="1">
    <nc r="A1" t="inlineStr">
      <is>
        <t>foo</t>
        <phoneticPr fontId="0"/>
      </is>
    </nc>
  </rcc>
</revisions>
```

The log shows that the contents of a cell were revised, and the new cell contents is text containg "foo" as the string.

After inserting a new sheet, the header looks like this:

```
<headers xmlns="…" xmlns:r="…" guid="{7E1DAFA8-EF95-4865-8FE8-CC17B28635CF}"
  diskRevisions="1" revisionId="2" version="3">
  <header guid="{A84A6777-8908-4CB9-9EB6-625CEFF419D3}"
    dateTime="2006-07-14T13:42:54" maxSheetId="4"
    userName="UserName" r:id="rId1">
    <sheetIdMap count="3">
      <sheetId val="1"/>
      <sheetId val="2"/>
      <sheetId val="3"/>
    </sheetIdMap>
  </header>
  <header guid="{CFEA9B63-728B-4274-A346-0440E1573AB4}"
    dateTime="2006-07-14T13:44:40" maxSheetId="4" userName="UserName"
    r:id="rId2" minRId="1">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <sheetIdMap count="3">
        <sheetId val="1"/>
        <sheetId val="2"/>
        <sheetId val="3"/>
      </sheetIdMap>
    </header>
    <header guid="{7E1DAFA8-EF95-4865-8FE8-CC17B28635CF}"
      dateTime="2006-07-14T13:48:56" maxSheetId="5" userName="UserName"
      r:id="rId3" minRId="2">
      <sheetIdMap count="4">
        <sheetId val="1"/>
        <sheetId val="2"/>
        <sheetId val="3"/>
        <sheetId val="4"/>
      </sheetIdMap>
    </header>
  </headers>
```

You can see that the last, most recent, header entry shows an entry for the new sheet. The most recent log looks like this:

```
  <revisions xmlns="…" xmlns:r="…">
    <ris rId="2" sheetId="4" name="[shared example.xlsx]Sheet4"
      sheetPosition="3"/>
    <rcv guid="{841DBE00-ECD0-478E-893B-30CE5DABBEF5}" action="delete"/>
    <rcv guid="{841DBE00-ECD0-478E-893B-30CE5DABBEF5}" action="add"/>
  </revisions>
```

This shows the new sheet, sheetId 4, is added to the workbook. The custom view (rcv) for the user is updated as a new sheet was added.

end example]

18.11.1.1 header (Header)

This element is essentially a table that contains metadata about a list of specific changes that have taken place for this workbook. It lists when the changes were made, who made them, and the relationship IDs so that the log detailing the specific change can be found. If tracking changes, or sharing workbooks, are enabled, then changes are persisted on the Save event, or at a specified time interval. A header is created for each set of changes.

Attributes	Description
dateTime (Date Time)	The date and time when this set of revisions was saved. [*Note*: This can happen when the user explicitly saves, or the save can occur due to a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	time interval, specified in the spreadsheet application, elapsing. *end note*] The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
guid (GUID)	A globally unique identifier for this set of revisions. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This is the ID that is used to find the corresponding log record of the changes made for this header. Use the corresponding relationship expressed in the revisionHeaders part to locate the log record that lists the specific changes. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
maxRId (Max Revision Id)	The highest revision Id that belongs to this header. [*Note*: This can be used when, given a revision ID, the spreadsheet application needs to determine which revision log to access. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
maxSheetId (Last Sheet Id)	Internal identifier of the next available sheet in this workbook. The numbering here is the index of the next available sheet in the workbook in a 1-based index system. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
minRId (Minimum Revision Id)	The lowest revision id that belongs to this header. [*Note*: this can be used when, given a revision ID, the spreadsheet application needs to determine which revision log to access. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
userName (User Name)	A string representing the name of the user making the revision.. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionHeader) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.11.1.2 headers (Revision Headers)

This element represents the list of revision headers.

This section contains many references to history, versions, and revisions, and it is helpful to clarify the relationships here. In general, a series of changes (revisions) can be made to a spreadsheet. When a batch of those revisions is saved to disk, the version number of the spreadsheet is incremented. The batch of changes is saved to the revision history, which is persisted on disk with the file in the form of different log files and headers.

There are some attributes that deal with history which might seem redundant (such as `diskRevisions`, and `history`, among others) - these are there for backwards compatibility with older versions of spreadsheet applications and do not need to be used for creating new files.

Attributes	Description
diskRevisions (Disk Revisions)	A Boolean value indicating that this shared workbook file contains revisions. `True` when the workbook does have revisions, `false` otherwise. [*Note*: this attribute is used for backwards compatibility. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
exclusive (Exclusive Mode)	A Boolean value indicating that this shared workbook is in exclusive mode. A workbook is in exclusive mode when a user has a lock on it for appending revisions to the file. [*Note*: This is used for backwards compatibility with older spreadsheet applications. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
guid (Last Revision GUID)	The globally unique identifier of the last set of revisions. This shall match the GUID for the most recent header. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
history (History)	A Boolean value indicating that this shared workbook maintains a revision history. True if a history is maintained, false otherwise. [*Note*: This is used for backwards compatibility with older spreadsheet applications. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
keepChangeHistory (Keep Change	A Boolean value indicating whether the revision history should be kept for this shared

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
History)	workbook. True if the history should be kept, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lastGuid (Last GUID)	Unique identifier of the last set of revisions that was saved into the file. The spreadsheet application can have certain modes, such as a timed save mode, where the application doesn't do a full save, but instead just appends the most recent revision records. In cases like this, for a new user that opens such a file while it is being edited, the file that was loaded from disk only have the changes that were saved during a full save. To get the current state of the file which includes edits by other users, the spreadsheet application would need to apply all the revisions from `lastGuid` to `guid`. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
preserveHistory (Preserve History)	An integer representing the number of days the spreadsheet application shall keep the change history for this workbook. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
protected (Protected)	A Boolean value indicating whether the change tracking in this shared workbook can be removed. True if the tracking can be removed, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
revisionId (Revision Id)	The current revision number of this shared workbook. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
shared (Shared Workbook)	A Boolean value indicating that this workbook is shared. True when the workbook is shared, false otherwise. [*Note*: This is used for backwards compatibility with older spreadsheet applications. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
trackRevisions (Track Revisions)	A Boolean value indicating that revisions are tracked in this shared workbook. True when revisions are tracked, false otherwise. [*Note*: This is used for backwards compatibility with older spreadsheet applications. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
version (Version)	An integer representing the current version of this shared workbook. The integer should begin counting from 1 for the first version. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionHeaders) is located in §A.2. *end note*]

18.11.1.3 nc (New Cell Data)

This element represents new cell data that was added to the worksheet.

For most spreadsheet application purposes, only the data type and reference need to be used for revision tracking purposes. The rest of the cell properties can be written out, but are not necessarily needed as they can be recorded in other areas of the spreadsheet. For instance the <rfmt> element can be used to record style information instead of the `S (style index)` attribute.

Attributes	Description
cm (Cell Metadata Index)	The zero-based index of the cell metadata record associated with this cell. Metadata information is found in the Metadata Part. Cell metadata is extra information stored at the cell level, and is attached to the cell (travels through moves, copy / paste, clear, etc). Cell metadata is not accessible via formula reference. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ph (Show Phonetic)	A Boolean value indicating if the spreadsheet application should show phonetic information. Phonetic information is displayed in the same cell across the top of the cell and serves as a 'hint' which indicates how the text should be pronounced. This should only be used for East Asian languages. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Reference)	An A1 style reference to the location of this cell The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
s (Style Index)	The index of this cell's style. Style records are stored in the Styles Part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
t (Cell Data Type)	An enumeration representing the cell's data type. The possible values for this attribute are defined by the ST_CellType simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§18.18.11).
vm (Value Metadata Index)	The zero-based index of the value metadata record associated with this cell's value. Metadata records are stored in the Metadata Part. Value metadata is extra information stored at the cell level, but associated with the value rather than the cell itself. Value metadata is accessible via formula reference. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cell) is located in §A.2. *end note*]

18.11.1.4 ndxf (New Formatting Information)

This element represents new differential formatting information for this cell. This formatting is applied to the existing formatting of the cell.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dxf) is located in §A.2. *end note*]

18.11.1.5 oc (Old Cell Data)

This element represents old cell data. Old cell data is data that was previously stored in the cell.

For most spreadsheet application purposes, only the data type and reference need to be used for revision tracking purposes. The rest of the cell properties can be written out, but are not necessarily needed as they can be recorded in other areas of the spreadsheet. For instance the <rfmt> element can be used to record style information instead of the `S (style index)` attribute.

Attributes	Description
cm (Cell Metadata Index)	The zero-based index of the cell metadata record associated with this cell. Metadata information is found in the Metadata Part. Cell metadata is extra information stored at the cell level, and is attached to the cell (travels through moves, copy / paste, clear, etc). Cell metadata is not accessible via formula reference. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ph (Show Phonetic)	A Boolean value indicating if the spreadsheet application should show phonetic information. Phonetic information is displayed in the same cell across the top of the cell and serves as a 'hint' which indicates how the text should be pronounced. This should only be used for East Asian languages. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Reference)	An A1 style reference to the location of this cell

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
s (Style Index)	The index of this cell's style. Style records are stored in the Styles Part. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
t (Cell Data Type)	An enumeration representing the cell's data type. The possible values for this attribute are defined by the ST_CellType simple type (§18.18.11).
vm (Value Metadata Index)	The zero-based index of the value metadata record associated with this cell's value. Metadata records are stored in the Metadata Part. Value metadata is extra information stored at the cell level, but associated with the value rather than the cell itself. Value metadata is accessible via formula reference. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cell) is located in §A.2. *end note*]

18.11.1.6 odxf (Old Formatting Information)

This element represents the old differential formatting information for this cell. Old differential formatting is differential formatting that was previously applied to the cell.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Dxf) is located in §A.2. *end note*]

18.11.1.7 oldFormula (Old Formula)

This element represents the old formula for a defined name in this cell. This is only used for named cells. Formulas that are entered in a cell with no name are represented by the formula element `<f>`.

The possible values for this element are defined by the ST_Formula simple type (§18.18.35).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Formula) is located in §A.2. *end note*]

18.11.1.8 raf (Revision AutoFormat)

This element represents a revision record of auto formatting change information for a table.

Attributes	Description
applyAlignmentFormats (Apply Alignment Formats)	If `true` apply legacy table autoformat alignment properties. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
applyBorderFormats (Apply Border Formats)	If `true` apply legacy table autoformat border properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyFontFormats (Apply Font Formats)	If `true` apply legacy table autoformat font properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyNumberFormats (Apply Number Formats)	If `true` apply legacy table autoformat number format properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyPatternFormats (Apply Pattern Formats)	If `true` apply legacy table autoformat pattern properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyWidthHeightFormats (Apply Width / Height Formats)	If `true` apply legacy table autoformat width/height properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoFormatId (Auto Format Id)	Identifies which legacy table autoformat to apply. Annex D contains a listing of the supported PivotTable AutoFormats, example formatting, and a sample workbook with each of those AutoFormats applied. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ref (Reference)	A-1 style reference to the location where the formatting was applied The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionAutoFormatting) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.11.1.9 rcc (Revision Cell Change)

This element stores information about the contents of the cell that was replaced.

Attributes	Description
dxf (Formatting)	A Boolean flag indicating that there was a differential formatting change for this cell - true if there was a formatting change, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
endOfListFormula Update (End of List Formula Update)	A Boolean flag indicating that indicates that the formula used at the end of a list has been updated. True if the formula was updated, false otherwise. List in this context does not mean table, rather it refers to the feature where the spreadsheet application automatically creates an internal structure for making data input more consistent on adjacent rows or columns. For instance, if 3 cells in a row are entered with the same format, then when entering data into the 4th adjacent cell, the spreadsheet application might automatically apply that same format. In this case, those cells are treated as a list. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
numFmtId (Number Format Id)	Zero-based index of the number format (Fmt) record used by this cell format (XF). The possible values for this attribute are defined by the ST_NumFmtId simple type (§18.18.47).
odxf (Old Formatting)	Flag indicating that there is old formatting information available for this cell. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
oldPh (Old Phonetic Text)	A Boolean flag indicating whether there is old phonetic text information available. True when there is old phonetic text information available, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
oldQuotePrefix (Old Quote Prefix)	A Boolean value indicating if a single quote prefix is was used on this cell previously. Single quote prefixes are used to cause a formula to be evaluated as a string. True if a single quote prefix was used previously, false otherwise The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ph (Phonetic Text)	A Boolean flag indicating whether this cell contains phonetic text or not. True when the cell contains phonetic text, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
quotePrefix (Quote Prefix)	A Boolean value indicating if a single quote prefix is used. Single quote prefixes are used to cause a formula to be evaluated as a string. True if a single quote prefix is used, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
s (Style Revision)	Flag indicating that formatting change for this cell affected the cell's style. (Only applicable for Undo operations) The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sId (Sheet Id)	Internal identifier of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xfDxf (Row Column Formatting Change)	Flag indicating that the formatting change had an effect on the formatting of the entire row or column that this cell belongs to. (Only applicable for Undo operations). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionCellChange) is located in §A.2. *end note*]

18.11.1.10 rcft (Revision Merge Conflict)

This element represents a revision record which indicates that there was a merge conflict.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionConflict) is located in §A.2. *end note*]

18.11.1.11 rcmt (Revision Cell Comment)

This element represents a revision record of a cell comment change.

Attributes	Description
action (User Action)	An enumeration identifying what kind of an operation the user performed on the comment. The possible values for this attribute are defined by the ST_RevisionAction simple type (§18.18.65).
alwaysShow (Always Show Comment)	A Boolean value indicating that the user has set this comment to always be visible. True if the comment is set to always be visible, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
author (Author)	A string representing the name of the author who changed this comment. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
cell (Cell)	An A-1 style reference to the cell where the comment was changed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
guid (GUID)	A globally unique identifier of this comment. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
hiddenColumn (Hidden Column)	A Boolean value indicating that the comment belongs to a cell in a hidden column. True if the comment is in a hidden column, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hiddenRow (Comment In Hidden Row)	A Boolean value indicating that the comment belongs to a cell in a hidden row. True if the comment is in a hidden row, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
newLength (New Comment Length)	Length of the comment text added in this revision. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
old (Old Comment)	An ignorable Boolean value used for backwards compatibility that indicates that the original comment was created by a legacy spreadsheet application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
oldLength (Original Comment Length)	Length of the comment before this revision was made. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionComment) is located in §A.2. *end note*]

18.11.1.12 rcv (Revision Custom View)

This element represents a revision record of adding or removing a custom view to the workbook

Attributes	Description
action (User Action)	An enumeration representing the action that the user performed. The possible values for this attribute are defined by the ST_RevisionAction simple type (§18.18.65).
guid (GUID)	A globally unique identifier of the custom view. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionCustomView) is located in §A.2. *end note*]

18.11.1.13 rdn (Revision Defined Name)

This element represents a revision record of a defined name change.

Attributes	Description
comment (Name Comment)	A string representing a comment about the defined name. This comment can be shown by the spreadsheet application in a names management UI so that users have more information about what the defined name is used for. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
customMenu (New Custom Menu)	A string representing the new custom menu text The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
customView (Custom View)	A Boolean flag indicating that this named range belongs to a custom view The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
description (Description)	A string representing the new description text for the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
function (Function)	A Boolean value indicating that the defined name refers to a function. True if the defined name is a function, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
functionGroupId (Function Group Id)	Represents the new function group id. Function group ids are used to help classify functions. For instance, functions in the same

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	group can be searched or selected easily from the spreadsheet applications UI. For instance, filtering the list of all functions to allow the user to choose from functions used for financial data. The following group ids should be used: ID Function group 1 Financial 2 Date and Time 3 Math and Trig 4 Statistical 5 Lookup & Reference 6 Database 7 Text 8 Logical 9 Information 10 Commands 11 Customizing 12 Macro Control 13 DDE/External 14 User Defined 15 Engineering 14 Cube The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
help (New Help Topic)	A string representing the new help topic text. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
hidden (Named Range Hidden)	A Boolean value indicating whether the named range is now hidden. Hidden refers to whether the defined name is of a 'hidden' type. This applies to things like a custom filter on a cell, it has a name, but is hidden and so is not visible in any name management UI. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
localSheetId (Local Name Sheet Id)	An integer representing the id of the sheet to which this defined name belongs. This shall be used local defined names only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Name)	A string representing the name for this defined name. The possible values for this attribute are defined by the ST_Xstring simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§22.9.2.19).
oldComment (Old Name Comment)	A string representing the old comment about the defined name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oldCustomMenu (Old Custom Menu Text)	A string representing the old custom menu text The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oldDescription (Old Description)	A string representing the old description text The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oldFunction (Old Function)	A Boolean flag indicating that the old name was a function The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
oldFunctionGroupId (Old Function Group Id)	Old function group ID. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
oldHelp (Old Help Topic)	A string representing the old help topic text The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oldHidden (Old Hidden)	A Boolean flag indicating whether the named range was hidden Hidden refers to whether the defined name is of a 'hidden' type. This applies to things like a custom filter on a cell, it has a name, but is hidden and so is not visible in any name management UI. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
oldShortcutKey (Old Short Cut Key)	Old keyboard shortcut. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
oldStatusBar (Old Status Bar)	A string representing the old status bar text The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ra (Revision Undo	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Rejected)	being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
shortcutKey (Shortcut Key)	Represents the new keyboard shortcut. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
statusBar (Status Bar)	A string representing the new status bar text. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionDefinedName) is located in §A.2. *end note*]

18.11.1.14 reviewed (Reviewed)

This element represents an identifier of a single reviewed revision. A reviewed revision, is a revision that has been reviewed via the spreadsheet application's track changes feature, has been accepted, and has been saved.

Attributes	Description
rId (revision Id)	ID of a reviewed revision. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Reviewed) is located in §A.2. *end note*]

18.11.1.15 reviewedList (Reviewed List)

This element maintains a list of reviewed revisions.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
count (Reviewed Revisions Count)	Number of reviewed revisions. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ReviewedRevisions) is located in §A.2. *end note*]

18.11.1.16 revisions (Revisions)

This element represents the root node of a list of revisions made in this shared workbook. This root node shows up at the beginning of every log file that contains specific revisions made to the workbook.

When multiple users are sharing, and editing, a workbook at the same time, there can be conflicting changes. The spreadsheet application should have logic to resolve such conflicts, and the file format should only contain enough information so that the spreadsheet application can restore the workbook to the correct state after conflict resolution. Revisions can also be tracked by the spreadsheet application for review by the user at a later time (as opposed to only dealing with conflicts on a save event.) Some edits to workbooks are made as a result of this conflict resolution. So, there are cases where a revision is effectively undone by another user, and as a result that undoing is itself a revision that adds or changes data in the file. These operations are tracked by the ua and ra attributes of many different elements.

[*Example*:

Step 1:

User 1 inserts Column A. So the XML in the revision log would look like this:

```
<revisions xmlns="…" xmlns:r="…">
  <rrc rId="1" sId="1" ref="A1:A1048576" action="insertCol"/>
</revisions>
```

Step 2:

User 2 synchronizes the file to pick up that change, but then activates the Track Changes feature, and rejects that change. This effectively performs an undo on User 1's insertion. This is denoted in the file with the ua attribute meaning that this change happened as the result of an undo. The XML for the revision log would look like this:

```
<revisions xmlns="…" xmlns:r="…">
  <rrc rId="2" ua="1" sId="1" ref="A1:A1048576" action="deleteCol"/>
  <rcft rId="1" ua="1" sheetId="1"/>
</revisions>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Step 3:

User 1 enters "foo" in A1, and saves the file. A conflict resolution dialog is shown since User 2's version of the file removed the inserted Column A. User 1 chooses to accept their own changes. This undoes the change that User 2 made. So, in effect, it performed an undo on a previous undo operation. This is denoted in the file format by the ra attribute meaning that a the change occurred because a previous undo was undone. So the resulting XML for the newest log file looks like this:

```
<revisions xmlns="…" xmlns:r="…">
  <rrc rId="3" ua="1" ra="1" sId="1" ref="A1:A1048576" action="insertCol"/>
  <rcft rId="2" ua="1" sheetId="1"/>
  <rcc rId="4" sId="1">
    <nc r="A1" t="inlineStr">
      <is>
        <t>foo</t>
      </is>
    </nc>
  </rcc>
  <rcft rId="2" sheetId="1"/>
</revisions>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Revisions) is located in §A.2. *end note*]

18.11.1.17 rfmt (Revision Format)

This element represents a revision record of information about a formatting change.

Attributes	Description
length (Length)	The number of characters that were affected by a string change, counting from start. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
s (Style)	Flag indicating that this formatting change affected a cell's style. (Only applicable for Undo operations). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sqref (Sequence Of	A worksheet range to which this formatting was applied. [*Note*: For applications

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
References)	supporting the default grid size (see §18.17.5), full column and row references shall explicitly state the row and column components, e.g., "A1:A1048576" For column "A", and A1:XFD1 for row "1". Applications with larger grid sizes shall interpret these to mean "column A" and "row 1" respectively, for their larger grid size. *end note*] The possible values for this attribute are defined by the ST_Sqref simple type (§18.18.76).
start (Start index)	An integer representing an index showing which character a string change starts at within the string in the cell. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
xfDxf (Row or Column Formatting Change)	A Boolean flag indicating that this formatting change had an affect on the formatting of an entire row or column that an affected cell(s) belongs to. (Only applicable for Undo operations) The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionFormatting) is located in §A.2. *end note*]

18.11.1.18 ris (Revision Insert Sheet)

This element represents a revision record of a sheet that was inserted.

Attributes	Description
name (Sheet Name)	The name of the new sheet. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetPosition (Sheet Position)	An integer representing the zero based index of the new sheet in the sheets collection (§18.2.20). The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionInsertSheet) is located in §A.2. *end note*]

18.11.1.19 rm (Revision Cell Move)

This element represents a revision record on a cell(s) that moved.

Attributes	Description
destination (Destination)	New A1 style location of the cell(s) that were moved The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
source (Source)	The original A1 style location of the cell(s) that were moved The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sourceSheetId	An integer representing the internal id of the sheet where the cell(s) originally resided.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Source Sheet Id)	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionMove) is located in §A.2. *end note*]

18.11.1.20 rqt (Revision Query Table)

This element represents a revision record of a query table field change.

Attributes	Description
fieldId (Field Id)	ID of the specific query table field that was removed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ref (QueryTable Reference)	Location of the affected query table. The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionQueryTableField) is located in §A.2. *end note*]

18.11.1.21 rrc (Revision Row Column Insert Delete)

This element represents a revision record of a row/column insert/delete action.

Attributes	Description
action (User Action)	Indicates the action most recently performed on the row or column. The possible values for this attribute are defined by the ST_rwColActionType simple type (§18.18.66).
edge (Edge Deleted)	A Boolean flag indicating that a row or column is being deleted at the edge of a sorted

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	range (only applicable to a Delete Row/Column revision types). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
eol (End Of List)	A Boolean flag indicating that a row or a column is being inserted at the end of a list of data. List in this context does not mean table, rather it refers to the feature where the spreadsheet application automatically creates an internal structure for making data input more consistent on adjacent rows or columns. For instance, if 3 cells in a row are entered with the same format, then when entering data into the 4th adjacent cell, the spreadsheet application might automatically apply that same format. In this case, those cells are treated as a list. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ref (Reference)	A reference to the location of the rows/columns that were inserted or deleted. [*Note*: A reference to a whole column or row must include both the column and row components. For example, column A is referenced by "A1:A1048576", and row 1 is referenced by "A1:XFD1". However, because this attribute value is occurring in the context of an entire row or column insert, the column component of a row reference can be ignored, and the row component of a column reference can be ignored. *end note*] The possible values for this attribute are defined by the ST_Ref simple type (§18.18.62).
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionRowColumn) is located in §A.2. *end note*]

18.11.1.22 rsnm (Revision Sheet Name)

This element represents a revision record tracking the renaming a sheet.

Attributes	Description
newName (New Sheet Name)	A string representing the new sheet name The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
oldName (Old Sheet Name)	A string representing the old sheet name The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ra (Revision Undo Rejected)	A Boolean flag which indicates that this revision was due to a previous undo (ua) revision being rejected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rId (Revision Id)	An integer representing the number of this revision. This id shall apply to reviewable revision types only. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sheetId (Sheet Id)	An integer representing the internal id of the sheet on which the revision occurred. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
ua (Revision From Rejection)	A Boolean flag indicating that this revision occurred because another revision was rejected and therefore undone. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RevisionSheetRename) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.11.1.23 sheetId (Sheet Id)

This element represents a sheet that revision can take place on. Each sheet in the workbook should be represented by one of these elements, and each sheet has an id associated with it. Sheet ids are used to refer to sheets internally by the spreadsheet application.

Attributes	Description
val (Sheet Id)	An integer serving as a number by which to reference the sheet internally. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetId) is located in §A.2. *end note*]

18.11.1.24 sheetIdMap (Sheet Id Map)

This element represents a list of sheets and corresponding ids that are used for tracking revision records.

Attributes	Description
count (Sheet Count)	Number of sheets. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SheetIdMap) is located in §A.2. *end note*]

18.11.1.25 undo (Undo)

This element represents undo information for row/column deletion when there are functions in the spreadsheet that reference the deleted rows/columns. This element is not applicable for insert revisions.

Attributes	Description
array (Array Formula)	Flag indicating that the affected formula is an array formula. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
cs (Cross Sheet Move)	A Boolean flag indicating this was a cross-sheet move. True if it was a cross sheet move, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dn (Defined Name)	Identifies the named range that referenced the deleted cell range. Mutually exclusive

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	with the cell reference attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
dr (Range)	The range which was deleted that is referenced by the affected formula. The possible values for this attribute are defined by the ST_RefA simple type (§18.18.63).
exp (Expression)	Identifies the expression that should be adjusted in the corresponding formula. The possible values for this attribute are defined by the ST_FormulaExpression simple type (§18.18.36).
index (Index)	Index of the expression within the corresponding formula that was affected by this change. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
nf (Defined Name Formula)	A Boolean flag indicating that the corresponding formula is part of a defined name. True if this formula is part of a defined name, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
r (Cell Reference)	Location of the cell whose formula referenced the deleted cell range. Mutually exclusive with the defined name attribute The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
ref3D (Reference 3D)	A Boolean flag indicating that the expression contained the sheet name in addition to the cell reference. True if it contained the sheet name, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sId (Sheet Id)	Internal Id of the worksheet that contained the formula that referenced the deleted cell range. Mutually exclusive with the defined name attribute. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
v (Value Needed)	A Boolean flag indicating the formula needs the actual value of the cell(s) it's referencing. True if the formula requires the value of the cell it references, false otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_UndoInfo) is located in §A.2. *end note*]

18.11.2 Shared Workbook User Data

This subclause specifies information about the users of a shared workbook.

18.11.2.1 userInfo (User Information)

This element represents a user, and it stores information about a specific user as it relates to revisions.

Attributes	Description
dateTime (Date Time)	Date and time when this user opened the shared workbook. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
guid (User Revisions GUID)	A globally unique identifier identifying the last set of revisions that this uses is synchronized to. This attribute can be used by the spreadsheet application to ensure that revisions this user depends on aren't deleted. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
id (User Id)	An integer representing an internal user id for this user. This number can be negative. The possible values for this attribute are defined by the W3C XML Schema int datatype.
name (User Name)	Display name for this user [*Note*: User name strings should not be longer than 54 characters. *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SharedUser) is located in §A.2. *end note*]

18.11.2.2 users (User List)

This element represents a list of users who currently have this shared workbook open. This list does not include any users who have the workbook open in Read-Only mode.

Attributes	Description
count (Active User Count)	Number of users who currently have this shared workbook open.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Users) is located in §A.2. *end note*]

18.12 QueryTable Data

Query tables are 2 dimensional tables of data bound to an external query of some kind. [*Example*: A query table could for example show specific data from a text file, from a web query, or from a database query. *end example*]

[*Example*:

Data connectivity can use a number of different technologies. The following spreadsheetML fragment is one an example of a query table connected to a database:

```
<queryTable xmlns="…" name="Northwind Orders" rowNumbers="1"
  growShrinkType="overwriteClear" connectionId="1" autoFormatId="16"
  applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="0"
  applyPatternFormats="0" applyAlignmentFormats="0" applyWidthHeightFormats="0">
  <queryTableRefresh nextId="15">
    <queryTableFields count="12">
      <queryTableField id="1" name="OrderID" tableColumnId="1"/>
      <queryTableField id="2" name="CustomerID" tableColumnId="2"/>
      <queryTableField id="3" name="EmployeeID" tableColumnId="3"/>
      <queryTableField id="4" name="OrderDate" tableColumnId="4"/>
      <queryTableField id="5" name="RequiredDate" tableColumnId="5"/>
      <queryTableField id="6" name="ShippedDate" tableColumnId="6"/>
      <queryTableField id="7" name="ShipName" tableColumnId="7"/>
      <queryTableField id="8" name="ShipAddress" tableColumnId="8"/>
      <queryTableField id="9" name="ShipCity" tableColumnId="9"/>
      <queryTableField id="10" name="ShipRegion" tableColumnId="10"/>
      <queryTableField id="11" name="ShipPostalCode" tableColumnId="11"/>
      <queryTableField id="12" name="ShipCountry" tableColumnId="12"/>
    </queryTableFields>
  </queryTableRefresh>
</queryTable>
```

end example]

[*Example*: And here's an example of the SpreadsheetML fragment defining a query table connected to a text import:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<queryTable xmlns="…" name="data in text" connectionId="1" autoFormatId="16"
  applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="1"
  applyPatternFormats="1" applyAlignmentFormats="0"
  applyWidthHeightFormats="0"/>
```

Elsewhere in the spreadsheetML file, a connection element is defined with the name "Northwind Orders" that describes how to connect to the appropriate database to refresh data for the query table. *end example*]

18.12.1 deletedField (Deleted Field)

This element specifies a field that has been deleted from the query table.

[*Example*:

```
<queryTableDeletedFields count="2">
  <deletedField name="ShipVia"/>
  <deletedField name="Freight"/>
</queryTableDeletedFields>
```

end example]

Attributes	Description
name (Deleted Fields Name)	Specifies the name of the deleted field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DeletedField) is located in §A.2. *end note*]

18.12.2 queryTable (Query Table)

This element specifies all the relevant properties for a query table, one query table element is stored for each query table object in the spreadsheetML document.

Attributes	Description
adjustColumnWidth (Adjust Column Width On Refresh)	Specifies whether to automatically adjust column widths on refresh to fit the data retrieved. `true` if column widths should be adjusted. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyAlignmentFormats (Apply Alignment Formats)	If `true` apply legacy table autoformat alignment properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
applyBorderFormats (Apply Border Formats)	If `true` apply legacy table autoformat border properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyFontFormats (Apply Font Formats)	If `true` apply legacy table autoformat font properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyNumberFormats (Apply Number Formats)	If `true` apply legacy table autoformat number format properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyPatternFormats (Apply Pattern Formats)	If `true` apply legacy table autoformat pattern properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
applyWidthHeightFormats (Apply Width / Height Formats)	If `true` apply legacy table autoformat width/height properties. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoFormatId (Auto Format Id)	Identifies which legacy table autoformat to apply. Annex D contains a listing of the supported PivotTable AutoFormats, example formatting, and a sample workbook with each of those AutoFormats applied. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
backgroundRefresh (Background Refresh)	Specifies whether or not the query table shall try to refresh data in the background. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
connectionId (Connection Id)	Specifies the ID number of the external data connection to use to refresh data in the query table. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
disableEdit (Disable Edit)	Specifies whether the connection element used with this query table shall be editable. If `true`, then the connection is not editable. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
disableRefresh (Disable Refresh)	Specifies whether the query table shall be refreshable. If `true`, then then query table is not refreshable. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fillFormulas (Fill Adjacent Formulas)	Specifies whether or not formulas in columns adjacent to the query table should be filled down whenever the query table is refreshed. This is helpful since the number of rows returned by a query table refresh operation can vary. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
firstBackgroundRefresh (First Background Refresh)	Specifies whether or not data has ever been refreshed for this query table. If the very first background data refresh had not completed at the time the file was saved, this attribute is set to `true`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
growShrinkType (Grow Shrink Type)	Specifies the type of behavior expected for dealing with a variable number of rows of data in the query table between refresh operations. The meaning of the possible values of this attribute {`insertClear`, `insertDelete`, `overwriteClear`} are explained in detail in the definition of the simple type. The possible values for this attribute are defined by the ST_GrowShrinkType simple type (§18.18.39).
headers (First Row Column Titles)	Specifies whether or not the query table has first row with column titles. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
intermediate (Intermediate)	Specifies whether this query table is in an intermediate state, having been defined but not fully formed and populated with data. In this state, fields and ranges of the query table can be unknown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (QueryTable Name)	Specifies the name of the query table. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
preserveFormattin	Specifies whether the application should try to preserve formatting in the query table

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
g (Preserve Formatting On Refresh)	and copy this formatting to any new rows of data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
refreshOnLoad (Refresh On Load)	Specifies whether the query table shall refresh its data automatically when the spreadsheetML document is loaded or opened. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
removeDataOnSave (Remove Data On Save)	Specifies whether the query table shall remove all data from the worksheet before the spreadsheetML document is saved. This is very helpful for situations where people who have different permissions to view data want to share the same spreadsheetML document. All data from the last user is removed, and new users re-query the external data sources with their own credentials. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowNumbers (Row Numbers)	Specifies whether the query table shall include a first column of row numbers. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryTable) is located in §A.2. *end note*]

18.12.3 queryTableDeletedFields (Deleted Fields)

This element is the collection for deletedField (§18.12.1) elements, each of which represents a column or field that has been deleted from the query table.

[*Example*:

```
<queryTableDeletedFields count="2">
  <deletedField name="ShipVia"/>
  <deletedField name="Freight"/>
</queryTableDeletedFields>
```

end example]

Attributes	Description
count (Deleted Fields Count)	Specifies how many deleted fields there are. The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryTableDeletedFields) is located in §A.2. *end note*]

18.12.4 queryTableField (QueryTable Field)

This element holds the properties related to a specific field or column in a query table.

Attributes	Description
clipped (Clipped Column)	Specifies whether this field/column is currently clipped and thus not visible in the worksheet. [*Note*: this state might occur for example when a query table is defined near the edge of a worksheet or other object in the spreadsheet that can't be overwritten with external data. In this case some of the fields are displayed, but not all of them. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dataBound (Data Bound Column)	Specifies whether this column is a user-defined column or comes from the external data query. User defined columns shall be preserved during data refresh operations. User-defined columns are only supported on query tables that are attached to table objects. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fillFormulas (Fill This Formula On Refresh)	Specifies whether the formula in this field/column should be filled down on data refresh. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Field Id)	Specifies the unique identifier of the query table field. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Name)	Specifies the unique name of the query table field. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
rowNumbers (Row Numbers)	`true` if this column contains the row numbers for the records returned. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tableColumnId (Table Column Id)	Specifies the unique identifier for the table column if the query table is attached to a table object rather than just a range in the sheet.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryTableField) is located in §A.2. *end note*]

18.12.5 queryTableFields (Query table fields)

This element is the collection for queryTableField elements.

Attributes	Description
count (Column Count)	Specifies the number of columns there are in this query table. Includes both query-defined and user-defined columns, but not deleted columns. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryTableFields) is located in §A.2. *end note*]

18.12.6 queryTableRefresh (QueryTable Refresh Information)

This element contains information related to refreshing the query table.

Attributes	Description
fieldIdWrapped (Next Field Id Wrapped)	Whether or not the idFieldNext value wrapped around. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
headersInLastRefresh (Headers In Last Refresh)	Whether or not the Query Table had titles last refresh. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minimumVersion (Minimum Refresh Version)	For backward compatibility with legacy versions of spreadsheet applications, this attribute specifies the minimum version of the application that is expected to correctly refresh the data in the query table without any problems. If this attribute is specified, an earlier version of a spreadsheet application should alert the user to the potential incompatibilities when a refresh is attempted. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
nextId (Next field id)	Specifies the next unique queryTableField (§18.12.4) id number available for assignment. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
preserveSortFilterLayout (Preserve Sort & Filter Layout)	Specifies whether sorting, autofilter, layout, and table block formatting should be preserved for this query table across data refresh operations. If this attribute is set to `false`, the query table might be more or less recreated from scratch when data is refreshed. In this case, all user deleted or rearranged columns, user inserted columns that aren't bound to external data, and table column formatting are discarded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
unboundColumnsLeft (Columns Left)	Specifies the number of extra columns included at the left end of the field array that aren't bound to external data. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
unboundColumnsRight (Columns Right)	Specifies the number of extra columns included at the right end of the Table that aren't bound to external data. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_QueryTableRefresh) is located in §A.2. *end note*]

18.13 External Data Connections

SpreadsheetML allows for the definition of top level data connection objects that describe how to retrieve data from external sources. These connection objects are independent of the constructs in the spreadsheet application that display data such as tables, PivotTables, etc.

Some information about a connection is considered part of the connection's definition. Other information is not inherently part of the connection, but it describes the way the connection is to be used by the containing workbook. Note that in many cases, the spreadsheet application does not need knowledge of the command syntax for the external data source (e.g., database query language), and simply stores a command string that was created by a data provider API (e.g., an ODBC driver).

A connection's definition can be established in a standalone connection file for easier sharing and reuse, but this reference documentation deals with the XML representation for external data connections that is directly

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

embedded within a SpreadsheetML document. This embedded representation is expected whenever external data is used, and ensures portability of the document and continued operation of the external query in the most cases.

18.13.1 connection (Connection)

This element contains both the definition of how to get at an external data source as well as information describing how the connection is used within the workbook. Specific constructs in a worksheet, such as OLAP formulas, QueryTables, or PivotTables make use of information in the connection to retrieve or refresh data based on default events or the user's explicit request.

Attributes	Description
background (Background Refresh)	Indicates whether the connection can be refreshed in the background (asynchronously). `true` if preferred usage of the connection is to refresh asynchronously in the background; `false` if preferred usage of the connection is to refresh synchronously in the foreground. This flag should be intentionally ignored in specific cases. [*Example*: An example of when the flag would be ignored is in the case of a connection to OLAP data on Microsoft SQL Server Analysis Services, where the connection is used by both a PivotTable and also by CUBE functions within the workbook. That connection will always be refreshed synchronously by the PivotTable and will always be refreshed asynchronously by the CUBE functions. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
credentials (Reconnection Method)	Specifies the authentication method to be used when establishing (or re-establishing) the connection. The possible values for this attribute are defined by the ST_CredMethod simple type (§18.18.16).
deleted (Deleted Connection)	Indicates whether the associated workbook connection has been deleted. `true` if the connection has been deleted; otherwise, `false`. Deleted connections contain only the attributes name and deleted=`true`, all other information is removed from the SpreadsheetML file. If a new connection is created with the same name as a deleted connection, then the deleted connection is overwritten by the new connection. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
description (Connection Description)	Specifies the user description for this connection. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
id (Connection Id)	Specifies The unique identifier of this connection. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
interval (Automatic Refresh Interval)	Specifies the number of minutes between automatic refreshes of the connection. When this attribute is not present, the connection is not automatically refreshed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
keepAlive (Keep Connection Open)	`true` when the spreadsheet application should make efforts to keep the connection open. When `false`, the application should close the connection after retrieving the information. This corresponds to the MaintainConnection property of a PivotCache object. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
minRefreshableVersion (Minimum Version Required for Refresh)	For compatibility with legacy spreadsheet applications. This represents the minimum version # that is required to be able to correctly refresh the data connection. This attribute applies to connections that are used by a QueryTable. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
name (Connection Name)	Specifies the name of the connection. Each connection shall have a unique name. When a connection has been marked as deleted and then a new connection is added with the same name, the deleted connection is replaced with the new connection. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
new (New Connection)	`true` if the connection has not been refreshed for the first time; otherwise, `false`. This state can happen when the user saves the file before a query has finished returning. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
odcFile (Connection File)	Specifies the full path to external connection file from which this connection was created. If a connection fails during an attempt to refresh data, and reconnectionMethod=1, then the spreadsheet application will try again using information from the external connection file instead of the connection object embedded within the workbook. This is a benefit for data source and spreadsheetML document manageability. If the definition in the external connection file is changed (e.g., because of a database server name change), then the workbooks that made use of that connection will fail to connect

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	with their internal connection information, and reload the new connection information from this file. This attribute is cleared by the spreadsheet application when the user manually edits the connection definition within the workbook. Can be expressed in URI or system-specific file path notation. [*Note*: Applications can decide what forms of URI they support, and whether system-specific file path notations are supported. *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
onlyUseConnectionFile (Only Use Connection File)	Indicates whether the spreadsheet application should always and only use the connection information in the external connection file indicated by the odcFile attribute when the connection is refreshed. If `false`, then the spreadsheet application should follow the procedure indicated by the reconnectionMethod attribute described below. Applies to ODBC connections, and may be applied to custom data connections. This attribute is ignored for other types of connections. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
reconnectionMethod (Reconnection Method)	Specifies what the spreadsheet application should do when a connection fails. The values are as follows: `1 = As required`: On refresh use the existing connection information. If the existing information cannot be used to establish a connection, get updated connection information, if available from the external connection file. `2 = Always`: On every refresh get updated connection information from the external connection file, if available, and use that instead of the existing connection information. In this case the data refresh will fail if the external connection file is unavailable. `3 = Never`: Never get updated connection information from the external connection file even if it is available and even if the existing connection information cannot be used. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
refreshedVersion (Last Refresh Version)	For backward compatibility purposes, this attribute indicates the version of the spreadsheet application that last refreshed the connection.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This attribute applies to connections that are used by a query table. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.
refreshOnLoad (Refresh on Open)	`true` if this connection should be refreshed when opening the file; otherwise, `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saveData (Save Data)	`true` if the external data fetched over the connection to populate a table is to be saved with the workbook; otherwise, `false`. This exists for data security purposes - if no external data is saved in (or "cached") in the workbook, then current user credentials can be required every time to retrieve the relevant data, and people won't see the data the workbook author had last been using before saving the file. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
savePassword (Save Password)	`true` if the password is to be saved as part of the connection string; otherwise, False. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
singleSignOnId (SSO Id)	Identifier for Single Sign On (SSO) used for authentication between an intermediate spreadsheetML server and the external data source. [*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/cSSOReturnCodes_SV01001109.asp *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sourceFile (Source Database File)	Used when the external data source is file-based. When a connection to such a data source fails, the spreadsheet application attempts to connect directly to this file. Can be expressed in URI or system-specific file path notation. [*Note*: Applications can decide what forms of URI they support, and whether system-specific file path notations are supported. *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
type (Database Source Type)	Specifies the data source type. Values are as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	1. ODBC-based source 2. Custom data connection source 3. Custom data connection source 4. Web query 5. Custom data connection source 6. Text-based source 7. Custom data connection source 8. Custom data connection source~~DSP~~ Custom data connection source represents an application-defined connection technology. [*Note*: For example, Microsoft Office uses these values to represent DAO (2), application-defined connection file (3), OLE DB (5), ADO (7), and DSP (8) connections. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connection) is located in §A.2. *end note*]

18.13.2 connections (Connections)

This element exists when there are one or more connections in the workbook. It is a container for the individual connection objects.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connections) is located in §A.2. *end note*]

18.13.3 dbPr (Database Properties)

This element stores all properties associated with an ODBC or OLE DB external data connection.

[*Example*:

Data connectivity can use a number of different technologies. The following is one example XML fragment defining an OLE DB connection and the associated dbPr element:

```
<connection id="2"
  odcFile="C:\My Documents\My Data Sources\Northwind Orders.odc" keepAlive="1"
  name="Northwind Orders" description="northwind" type="5" refreshedVersion="3">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <dbPr connection="Provider=SQLOLEDB.1;Persist
      Security Info=True;Initial Catalog=Northwind;Data Source=dataserver1;Use
      Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation
      ID=LOCAL_MACHINE_NAME;Use Encryption for Data=False;Tag with column
      collation when possible=False"
      command="&quot;Northwind&quot;.&quot;dbo&quot;.&quot;Orders&quot;"
      commandType="3"/>
  </connection>
```

end example]

Attributes	Description

command
(Command Text)
	The string containing the database command to pass to the data provider that will interact with the external source in order to retrieve data. [*Note*: The SpreadsheetML application need not understand the command syntax. *end note*]
	
	[*Example*:
	Data connectivity can use a number of different technologies. The following is one example of an ODBC command string of commandType=2 (for a Microsoft SQL Server database):
	
	`command="SELECT Orders.OrderID, Orders.OrderDate, Orders.ShipName, Orders.ShipAddress, Orders.ShipCity, Orders.ShipRegion, Orders.ShipPostalCode, Orders.ShipCountry_x000d__x000a_FROM Northwind.dbo.Orders Orders_x000d__x000a_WHERE (Orders.ShipCountry=?)"`
	
	Some characters in this string have been escaped - for more information on the escaping scheme, please refer to the ST_Xstring simple type definition. *end example*]
	
	[*Note*: the "?" syntax in the string is something that the ODBC data provider is aware of and might replace with a parameter before execution. *end note*]
	
	[*Example*:
	Data connectivity can use a number of different technologies. The following is one example of an OLE DB command string of commandType=3 (for an Oracle database):
	
	`command=""TESTDB"."ShippersTable""`
	
	end example]
	
	[*Note*: Data connectivity can use a number of different technologies. A few examples of potential values stored in this attribute can be found at:
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsql_statements.asp
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsql_minimum_grammar.asp

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbusing_commands.asp *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
commandType (custom data source Command Type)	Specifies the custom data source command type. Values are passed to the custom data source provider. [*Example*: For the OLE DB custom data source provider, valid values are as follows: 1. Query specifies a cube name 2. Query specifies a SQL statement 3. Query specifies a table name 4. Query specifies that default information has been given, and it is up to the provider how to interpret. 5. Query is against a web based List Data Provider. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

connection (Connection String)	The string used to initiate a session with a data source. [*Note*: The SpreadsheetML application need not understand the connection syntax. *end note*]
	
	[*Example*: ODBC connection string to a database:
	
	`connection="DRIVER=SQL Server;SERVER=example_server;UID=example_useralias;APP=Microsoft Office 2007;WSID=user_alias;Trusted_Connection=Yes"`
	
	end example]
	
	[*Example*: of an OLE DB connection string to an Oracle database:
	
	`connection="Provider=OraOLEDB.Oracle.1;Password=example_password;Persist Security Info=True;User ID=example_useralias;Data Source=example_server;Extended Properties="""`
	
	end example]
	
	[*Note*: Data connectivity can use a number of different technologies. A few examples of potential values stored in this attribute can be found at:
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/dasdkodbcoverview.asp
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbcsql/od_odbc_d_4x4k.asp
	• http://msdn.microsoft.com/library/default.asp?url=/library/en-

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	us/ado270/htm/mdreforacleprovspec.asp *end note*] Connection strings syntaxes are specific to individual ODBC or custom data provider drivers. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
serverCommand (Command Text)	Specifies a second command text string that is persisted when PivotTable server-based page fields are in use. For ODBC connections, serverCommand is usually a broader query than command (no WHERE clause is present in the former). Based on these 2 commands, parameter UI can be populated and parameterized queries can be constructed. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DbPr) is located in §A.2. *end note*]

18.13.4 m (No Value)

This element is present when tables in a web query are missing.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableMissing) is located in §A.2. *end note*]

18.13.5 olapPr (OLAP Properties)

This element contains all the properties needed for an OLAP data connection. OLAP connections contain both the dbPr and olapPr child elements.

[*Example*:

Data connectivity can use a number of different technologies. The following is an example of a connection to an SAP BW OLAP data source:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<connection id="1" odcFile="C:\My Documents\My Data Sources\$INFOCUBE.odc"
  keepAlive="1" name="SAP demo cube" description="SAP DemoCube" type="5"
  refreshedVersion="3" background="1">
  <dbPr connection="Provider=MDrmSap.2;Data Source=BI2;User
    ID=TESTUSER;Location=TESTSERVERNAME;Cache Authentication=False;Encrypt
    Password=False;Integrated Security=&quot;&quot;;Mask Password=False;Persist
    Encrypted=False;Persist Security Info=True;Impersonation
    Level=Anonymous;Mode=Read;Protection Level=None;Extended
    Properties=&quot;SFC_CLIENT=800;&quot;;Initial Catalog=$INFOCUBE"
    command="$0D_DECU" commandType="1"/>
  <olapPr sendLocale="1" rowDrillCount="1000" serverFill="0"
    serverNumberFormat="0" serverFont="0" serverFontColor="0"/>
</connection>
```

end example]

[*Example*:

Data connectivity can use a number of different technologies. The following is an example of a connection to a Microsoft SQL Server Analysis Services OLAP data source:

```
<connection id="1"
  odcFile="C:\My Documents\My Data Sources\Adventure Works DW.odc" keepAlive="1"
  name="Adventure Works DW" type="5" refreshedVersion="3" background="1">
  <dbPr connection="Provider=MSOLAP.3;Cache Authentication=False;Persist
Security Info=True;Initial Catalog=Adventure Works
    DW;Data Source=DATASERVER1;Impersonation
    Level=Impersonate;Mode=ReadWrite;Protection Level=Pkt Privacy;Auto Synch
    Period=20000;Default Isolation Mode=0;Default MDX Visual Mode=0;MDX
    Compatibility=1;MDX Unique Name Style=0;Non Empty
    Threshold=0;SQLQueryMode=Calculated;Safety Options=2;Secured Cell
    Value=0;SOURCE_DSN_SUFFIX=&quot;Prompt=CompleteRequired;Window
    Handle=0x6A903CC;&quot;;SQL Compatibility=0;Compression Level=0;Real Time
    Olap=False;Packet Size=4096" command="Adventure Works" commandType="1"/>
  <olapPr sendLocale="1" rowDrillCount="1000"/>
</connection>
```

end example]

[*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/dasdkoledboverview.asp *end note*]

Attributes	Description
local (Local Cube)	Flag indicating whether we should get data from the local cube on refresh versus the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	original data source. `true` if a local cube has been created for OLAP data, and it should be used instead of the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
localConnection (Local Cube Connection)	Specifies a connection string to use when a local cube is available. This is used when local is set to `true`. [*Example*: `<olapPr local="true" localConnection="OLEDB;Provider=MSOLAP;Data Source=C:\Data\DataCube.cub" >` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
localRefresh (Local Refresh)	Flag indicating whether we should refresh the local cube from the original data source. When `true`, the original OLAP data source is queried each time the user explicitly refreshes the data in the application, and a new local cube is constructed from this query. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rowDrillCount (Drill Through Count)	Maximum number of drill-through rows to return when the user drills through an aggregate value in a PivotTable. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
sendLocale (Send Locale to OLAP)	When `true`, the spreadsheetML app should send the user interface locale ID to the OLAP provider to retrieve localized member names and properties, etc. When `false`, no locale ID is expected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
serverFill (OLAP Fill Formatting)	When `true` a PivotTable based on an OLAP source should format the data and aggregate cells in the PivotTable view using the background color from the OLAP source if this information is available. When `false`, OLAP server background fill colors are ignored, and standard formatting rules within the worksheet are followed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
serverFont (OLAP	When `true`, a PivotTable based on OLAP source should format the data and aggregate

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Server Font)	cells in the PivotTable view using the font from the OLAP source (e.g., Arial or Tahoma). When `false`, OLAP server fonts are ignored, and standard formatting rules within the worksheet are followed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
serverFontColor (OLAP Font Formatting)	When `true` a PivotTable based on OLAP source should format the data and aggregate cells in the PivotTable view using the font color from the OLAP source. When `false`, OLAP server font colors are ignored, and standard formatting rules within the worksheet are followed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
serverNumberFormat (OLAP Number Format)	When `true`, a PivotTable based on OLAP source should format the data and aggregate cells in the PivotTable view using the number format from the OLAP source. When `false`, OLAP server number formats are ignored, and standard formatting rules within the worksheet are followed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OlapPr) is located in §A.2. *end note*]

18.13.6 parameter (Parameter Properties)

This element stores properties about any parameters used with external data connections. Parameters are used to change the query executed externally and cause different data to be retrieved into the workbook. The type of parameter used – see ST_parameterType (§18.18.54) – determines whether the user is prompted for a value before data is refreshed, or the value is pulled from a cell in the workbook, or whether the same value should be used until explicitly changed in the data connection. Parameters are permitted for ODBC and web queries.

[*Example*:

Data connectivity can use a number of different technologies. The following is an example of XML defining a connection to a Microsoft Access database, with a parameter based on the value in cell C1 on the first sheet.

```
<connection id="1" name="Connection" type="1" refreshedVersion="2"
  background="1" saveData="1">
  <dbPr connection="DSN=MS Access
    Database;DBQ=C:\Desktop\db1.mdb;DefaultDir=C:\Desktop;DriverId=25;FIL=MS
    Access;MaxBufferSize=2048;PageTimeout=5;" command="SELECT Table1.Field1,
    Table1.Field2_x000d__x000a_FROM `C:\Desktop\db1`.Table1
    Table1_x000d__x000a_WHERE (Table1.Field2=?)"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <parameters count="1">
      <parameter name="user specified value" sqlType="4" parameterType="cell"
        cell="Sheet1!$C$1"/>
    </parameters>
  </connection>
```

end example]

Note that the command string in the dbPr element contains a "?" character. This character serves as a parameter marker.

[*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcstatement_parameters.asp *end note*]

Attributes	Description
boolean (Boolean)	Boolean value to use as the query parameter. Used only when parameterType = value. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
cell (Cell Reference)	Cell reference indicating which cell's value to use for the query parameter. Used only when parameterType = cell. [*Example*: `<Parameter parameterType="cell" cell="Sheet1!$C$1">` *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
double (Double)	Non-integer numeric value to use as the query parameter. Used only when parameterType = value. The possible values for this attribute are defined by the W3C XML Schema double datatype.
integer (Integer)	Integer value to use as the query parameter. Used when parameterType = value. The possible values for this attribute are defined by the W3C XML Schema int datatype.
name (Parameter Name)	The name of the parameter. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
parameterType (Parameter Type)	Type of parameter used. If the parameterType=value, then the value from boolean, double, integer, or string are used. In this case, it is expected that only one of {boolean,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	double, integer, or string} is specified. The possible values for this attribute are defined by the ST_ParameterType simple type (§18.18.54).
prompt (Parameter Prompt String)	Prompt string for the parameter. Presented to the spreadsheet user along with input UI to collect the parameter value before refreshing the external data. Used only when parameterType = prompt. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
refreshOnChange (Refresh on Change)	Flag indicating whether the query should automatically refresh when the contents of a cell that provides the parameter value changes. If `true`, then external data is refreshed using the new parameter value every time there's a change. If `false`, then external data is only refreshed when requested by the user, or some other event triggers refresh (e.g., workbook opened). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sqlType (SQL Data Type)	SQL data type of the parameter. This attribute is only supported for ODBC data sources. For more information, including supported data types, see ISO 9075-3:2008, Table 33 - Codes used for concise data types. The possible values for this attribute are defined by the W3C XML Schema int datatype.
string (String)	String value to use as the query parameter. Used only when parameterType = value. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Parameter) is located in §A.2. *end note*]

18.13.7 parameters (Query Parameters)

This element serves as a collection of parameters for an ODBC or web query.

Attributes	Description
count (Parameter Count)	The number of parameters used. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Parameters) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.13.8 s (Character Value)

This element is used to specify an HTML table to import by name. If the tables are not named, they shall be specified with the <x v="[index]"> syntax instead.

Attributes	Description
v (Value)	The name of the table to retrieve when the web query is refreshed. This corresponds to the string used for the id attribute of the HTML <table> tag. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_XStringElement) is located in §A.2. *end note*]

18.13.9 tables (Tables)

This element serves as the collection of tables to be returned via a web query data connection. Tables are then most commonly referenced by <x> via their indices (in order of the <Table> tags in the HTML page).

Attributes	Description
count (Count of Tables)	Number of tables to pull data from when refreshing from a web query. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tables) is located in §A.2. *end note*]

18.13.10 textField (Text Import Field Settings)

This element specifies field settings for text import.

Attributes	Description
position (Position)	The character position the field starts at for fixed-length fields. The index is 0-based. If this attribute does not exist, position=0 is assumed. Subsequent textField elements or carriage returns in the text stream serve to denote endpoints for text fields. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Field Type)	Specifies the field Type. When text is imported into cells in the worksheet, the data in the cells are converted to the field type defined here. Types can be specified by the user, or determined algorithmically via heuristics and text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	analysis. The possible values for this attribute are defined by the ST_ExternalConnectionType simple type (§18.18.27).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextField) is located in §A.2. *end note*]

18.13.11 textFields (Fields)

This element that denotes a set of fields to retrieve from a text file. Contains 1 or more textField elements.

Attributes	Description
count (Count of Fields)	Number of distinct fields to retrieve. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFields) is located in §A.2. *end note*]

18.13.12 textPr (Text Import Settings)

This element contains all of the text import settings.

[*Example*: Here's an example of the XML for a text connection:

```
<connection id="1" name="text data" type="6" refreshedVersion="3" background="1"
  saveData="1">
  <textPr prompt="0" characterSet="IBM437" sourceFile="C:\Desktop\text data.txt"
    delimiter="|">
    <textFields count="5">
      <textField/>
      <textField type="text" position="7"/>
      <textField type="text" position="28"/>
      <textField position="36"/>
      <textField type="text" position="41"/>
    </textFields>
  </textPr>
</connection>
```

example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>characterSet (Character Set)</td><td>Name of the character set associated with the text file. Values for this attribute are restricted to the names and aliases listed in the IANA CHARACTER SETS listing found at http://www.iana.org/assignments/character-sets.

[Note: When reading this value, if a system does not support a particular character set, the application is allowed to decide what is the best course of fallback action. end note]

If this attribute is not present then the codePage attribute are used.

The possible values for this attribute are defined by the W3C XML Schema string datatype.</td></tr>
<tr><td>comma (Comma is Delimiter)</td><td>Flag indicating whether to treat comma characters as field delimiters.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>consecutive (Consecutive Delimiters)</td><td>Flag indicating whether consecutive delimiters should be treated as just one delimiter. If this flag is <code>true</code> than it's possible or even likely that some rows will return more fields than others, and these fields will always fill cells in the worksheet from left to right.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>decimal (Decimal Separator)</td><td>The decimal separator character. This and the thousands attribute are used only when data in the text file contains decimal and thousands separators that are different from those used on the computer, due to a different language setting being used.

The following table shows the results when you import text into a spreadsheet application using various separators. Numeric results are displayed in the rightmost column.

<table>
<tr><th>System decimal separator</th><th>System thousands separator</th><th>Text file decimal separator value</th><th>Text file thousands Separator value</th><th>Text imported</th><th>Cell value (data type)</th></tr>
<tr><td>Period</td><td>Comma</td><td>Comma</td><td>Period</td><td>123.123,45</td><td>123,123.45 (numeric)</td></tr>
<tr><td>Period</td><td>Comma</td><td>Comma</td><td>Comma</td><td>123.123,45</td><td>123.123,45 (text)</td></tr>
<tr><td>Comma</td><td>Period</td><td>Comma</td><td>Period</td><td>123,123.45</td><td>123,123.45 (numeric)</td></tr>
<tr><td>Period</td><td>Comma</td><td>Period</td><td>Comma</td><td>123 123.45</td><td>123 123.45 (text)</td></tr>
<tr><td>Period</td><td>Comma</td><td>Period</td><td>Space</td><td>123 123.45</td><td>123,123.45 (numeric)</td></tr>
</table>

Strings values of this attribute are expected to be one character in length.</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
delimited (Delimited File)	`true` if the file is Tab or character delimited. `false` if the file should be parsed according to fixed length fields. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
delimiter (Custom Delimiter)	User-specified character to be treated as a field delimiter. Only single characters are supported. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
fileType (File Type)	Ignorable attribute with enum value that defined by ST_FileType. Determines the kind of character set to use during import. Only one of fileType and characterSet or codePage shall be specified for a textPr. The possible values for this attribute are defined by the ST_FileType simple type (§18.18.29).
firstRow (First Row)	Indicates at what row of the file to start the data import. All unsignedInt values are permitted, although it's possible that firstRow is higher than the number of rows in the text file, in which case no data is imported. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
prompt (Prompt for File Name)	Flag indicating whether the user wants to be prompted for the file name on refresh. If `false`, then the user is not prompted. If `true` or not present, then the user is prompted. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
qualifier (Qualifier)	Character used as the text string qualifier. The possible values for this attribute are defined by the ST_Qualifier simple type (§18.18.61).
semicolon (Semicolon is Delimiter)	Flag indicating whether to treat semicolon characters as field delimiters. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sourceFile (Source File Name)	Path to the text file to use to import external data. Can be expressed in URI or system-specific file path notation. [*Note*: Applications can decide what forms of URI they support, and whether system-

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	specific file path notations are supported. *end note*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
space (Space is Delimiter)	Flag indicating whether to treat space characters as field delimiters. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tab (Tab as Delimiter)	Flag indicating whether to treat tab characters as field delimiters. If `false`, then tabs will not be used as delimiters. If `true` or not present, then they are used as delimiters. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
thousands (Thousands Separator)	The thousands separator character. This and the decimal attribute are used only when data in the text file contains decimal and thousands separators that are different from those used on the computer, due to a different language setting being used. Please refer to the decimal attribute description above for a Table describing the behavior. Strings values of this attribute are expected to be one character in length. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextPr) is located in §A.2. *end note*]

18.13.13 webPr (Web Query Properties)

This element specifies the properties for a web query source. A web query will retrieve data from HTML tables, and can also supply HTTP "Get" parameters to be processed by the web server in generating the HTML by including the parameters and parameter elements.

Here's an example of a web query connection:

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<connection id="1" name="Connection" type="4" refreshedVersion="0"
  background="1" saveData="1">
  <webPr sourceData="1" parsePre="1" consecutive="1"
    url="http://ServerName/Image%20Library/Forms/AllItems.aspx" htmlTables="1">
    <tables count="1">
      <s v="contentthumbnail"/>
    </tables>
  </webPr>
</connection>
```

end example]

Attributes	Description
consecutive (Consecutive Delimiters)	Flag indicating whether consecutive delimiters should be treated as just one delimiter. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
editPage (Edit Query URL)	The URL of the user-facing web page showing the web query data. This URL is persisted in the case that sourceData="true" and url has been redirected to reference an XML file. Then the user-facing page can be shown in the UI, and the XML data can be retrieved behind the scenes. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
firstRow (Use First Row)	Flag indicating whether to parse all tables inside a PRE block with the same width settings as the first row. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
htmlFormat (HTML Formatting Handling)	How to handle formatting from the HTML source when bringing web query data into the worksheet. Relevant when sourceData is True. Values are as follows: 1. None - no formatting at all 2. RTF - honor just rich text formatting 3. All - honor all html formatting. The possible values for this attribute are defined by the ST_HtmlFmt simple type (§18.18.41).
htmlTables (HTML Tables Only)	Flag indicating whether web queries should only work on HTML tables. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
parsePre (Parse PRE)	Flag indicating whether data contained within HTML <PRE> tags in the web page is parsed into columns when you import the page into a query table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
post (Web Post)	Returns or sets the string used with the post method of inputting data into a web server to return data from a web query. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sourceData (Import XML Source Data)	Flag indicating that XML source data should be imported instead of the HTML table itself. Used when a web query exists to an HTML table with the following attribute. `<TABLE … o:WebQuerySourceHRef="http://…" …> … </TABLE>` The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
textDates (Dates as Text)	Flag indicating whether dates should be imported into cells in the worksheet as text rather than dates. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
url (URL)	URL to use to refresh external data. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
xl2000 (Refreshed in Excel 2000)	This flag exists for backward compatibility with older existing spreadsheet files, and is set to `true` if this web query was refreshed in a spreadsheet application newer than or equal to Microsoft Excel 2000. This is an optional attribute that can be ignored. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xl97 (Created in Excel 97)	This flag exists for backward compatibility with older existing spreadhseet files, and is set to `true` if this web query was created in Microsoft Excel 97. This is an optional attribute that can be ignored. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
xml (XML Source)	`true` if the web query source is XML (versus HTML), otherwise `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_WebPr) is located in §A.2. *end note*]

18.14 Supplementary Workbook Data

External links are used when linking the workbook to other workbooks or external data. The most frequent feature for linking a workbook to other workbooks is through the use of formulas. In this case the formula references a range or defined name in another workbook. Hyperlinks on cells and other spreadsheet objects are also considered an external link. Object-linking technologies are yet another technology used to link the workbook to another object. [*Example*: KParts or OLE. *end example*] Finally, Dynamic Data Exchange, or DDE, servers can be used to access external data. DDE servers are accessed through formulas in the workbook.

External links are saved with the target source in a relationship file so that external resources are easily discoverable in lightweight relationship XML rather than deep in the application's XML.

For a workbook consumer.xlsx that makes use of data in another workbook called data.xlsx, the following XML would exist in consumer.xlsx to describe the external link:

[*Example*:

```
<Relationships xmlns="…">
  <Relationship Id="rId1" Type="…/externalLinkPath" Target="data.xlsx"
    TargetMode="External"/>
</Relationships>
```

end example]

And the following XML would exist to describe cached data retrieved from the external workbook:

[*Example*:

```
<externalLink xmlns="…">
  <externalBook xmlns:r="…" r:id="rId1">
    <sheetNames>
      <sheetName val="Sheet1"/>
      <sheetName val="Sheet2"/>
      <sheetName val="Sheet3"/>
    </sheetNames>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <sheetDataSet>
      <sheetData sheetId="0"/>
      <sheetData sheetId="1"/>
      <sheetData sheetId="2">
        <row r="11">
          <cell r="B11">
            <v>47</v>
          </cell>
        </row>
        <row r="12">
          <cell r="B12">
            <v>19</v>
          </cell>
        </row>
        <row r="13">
          <cell r="B13">
            <v>38</v>
          </cell>
        </row>
      </sheetData>
    </sheetDataSet>
  </externalBook>
</externalLink>
```

end example]

The Supplementary Workbook Data section of SpreadsheetML is complimentary to the External Data Connnections (§18.13) section in maintaining all the information about external information that impacts the workbook.

18.14.1 cell (External Cell Data)

This element is used to store cached values from external sources such as other workbooks. Formulas from external cells are not stored in the consuming workbook. Also, for this context, the attribute t cannot have a value of `inlineStr`. Rich text is not supported in this context either.

Attributes	Description

r (Reference)	Describes the cell location in the external book.
	
	[*Example*:
	
	`<cell r="B12">`
	`  <v>74</v>`
	`</cell>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
t (Type)	Indicates the data type of the cell value. The possible values for this attribute are defined by the ST_CellType simple type (§18.18.11).
vm (Value Metadata)	The index of the cell's value metadata, if any exists. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalCell) is located in §A.2. *end note*]

18.14.2 ddeItem (DDE Item definition)

This element represents a DDE item.

Attributes	Description
advise (Advise)	Specifies whether the DDE server should notify the application when the external data changes. Default value is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (DDE Name)	Specifies the DDE item name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ole (Object Linking TechnologyE)	Set to `true` if this is item uses an object linking technology. Default value is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
preferPic (Data is an Image)	Set to `true` if data from this DDE item is an image format. Default value is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DdeItem) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.14.3 ddeItems (DDE Items Collection)

This element serves as a collection for ddeItem elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DdeItems) is located in §A.2. *end note*]

18.14.4 ddeLink (DDE Connection)

This element represents a connection to an external Dynamic Data Exchange (DDE) server. DDE is a method of sending data between applications using Windows messages according to a documented protocol that has been stable since about 1990.

The hierarchy of names defined by a DDE server is Application, Topics, and Items. Topics often correspond to units such as files or documents or database names, and Items refer to subsets of the data such as cell ranges, rows, fields, columns. DDE items can have multiple values as well.

[*Example*:

Data connectivity can use a number of different technologies.The following is just one example of a spreadsheetML fragment describing the product Microsoft Excel being used as a DDE server to provide data to the current spreadsheet document:

```
<ddeLink xmlns:r="…" ddeService="excel" ddeTopic="[ddesource.xls]Sheet1">
  <ddeItems>
    <ddeItem name="R1C1" advise="1"/>
    <ddeItem name="StdDocumentName" ole="1" advise="1"/>
  </ddeItems>
</ddeLink>
```

end example]

Attributes	Description
ddeService (Service name)	Service name (i.e., application name) for the DDE connection. This is a required attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
ddeTopic (Topic for DDE server)	Describes something for the DDE application to which the channel pertains— usually a document of that application. This is a required attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DdeLink) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.14.5 definedName (Defined Name)

This element contains information about a named range in an external workbook.

Attributes	Description
name (Defined Name)	The defined name. This attribute is required. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

refersTo (Refers To)	Name range definition string.
	
	[*Example*:
	
	`<definedNames>`
	`  <definedName name="namedrange"`
	`  refersTo="='Sheet1'!$D$5:$D$10"/>`
	`</definedNames>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

sheetId (Sheet Id)	The index of the worksheet that the named range applies to for named ranges that are scoped to a particular worksheet rather than the full workbook. This attribute is optional. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalDefinedName) is located in §A.2. *end note*]

18.14.6 definedNames (Named Links)

This element is a collection of the defined names associated with the supporting workbook.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalDefinedNames) is located in §A.2. *end note*]

18.14.7 externalBook (External Workbook)

This element represents an external workbook which is supplying data to the current workbook.

Attributes	Description
id (Relationship to supporting book file	Relationship ID that references a link in the relationships collection. The target attribute in the associated relationship will specify the worksheet XML file in the current

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
path) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	spreadsheetML document ZIP archive that makes use of this externalbook. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalBook) is located in §A.2. *end note*]

18.14.8 externalLink (External Reference)

This element is a container for specific types of external links.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalLink) is located in §A.2. *end note*]

18.14.9 oleItem (Object Link Item)

This element represents a single link within the object referenced by the parent element.

Attributes	Description
advise (Advise)	Set to `true` if the linked object should notify the application when the external data changes. Default value is `false`. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
icon (Icon)	Set to `true` if the linked object is represented by an icon. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
name (Object Name)	The linked object's name. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
preferPic (Object is an Image)	Set to `true` if the linked object is represented by an image. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleItem) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.14.10 oleItems (Object Link Items)

This element is a collection of items within the link specified by the parent element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleItems) is located in §A.2. *end note*]

18.14.11 oleLink (Generic Object Link Connection)

This element represents an external link to an embedded object, specified by a progID/object pair. The type of object link is determined by reading the target of the id attribute.

[*Example*: The following markup defines a reference to a linked object using Bonobo. The progId attribute contains the shared library that contains the widget. The r:id identifies the referenced Bonobo object.

```
<oleLink r:id="rb1" progId="OAFIID:Bonobo_Sample_Calculator">
  …
</oleLink>
```

The target of the relationship with ID rb1, defines the Bonobo object itself. This example shows a link to a sample Bonobo widget taken from the following article, which also provides an introduction to Bonobo – http://www.ibm.com/developerworks/webservices/library/co-bnbo2.html. *end example*]

[*Example*: The following markup defines a reference to a linked object using KParts. The progId attribute contains the shared library that contains the plugin. The r:id identifies the referenced KParts object.

```
<oleLink r:id="rKp1" progId="libhtmlvalidatorplugin">
  …
</oleLink>
```

The following XML, contained in the target of the relationship with ID rKp1, defines the KPart object, and will follow the kpartgui DTD:

```
<!DOCTYPE kpartgui SYSTEM "kpartgui.dtd">
<kpartgui library="libhtmlvalidatorplugin" name="htmlvalidatorplugin"
version="1" >
  <MenuBar>
    <Menu name="tools"><Text>&amp;Tools</Text>
      <Action name="validatewebpage"/>
    </Menu>
  </MenuBar>
</kpartgui>
```

This example is taken from the kde.org web site, and contains a tutorial on building the plugin referenced by the above markup – http://developer.kde.org/documentation/tutorials/dot/writing-plugins.html. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
id (Object Link Relationship) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Relationship ID that references a link in the relationships collection. The target attribute in the associated relationship will specify the external file name used for this oleLink. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
progId (Object Link Identifier)	The ID for the object link connection. [*Example*: For a KParts link, this would store the name of the appropriate KParts library. For an OLE link, this would store the ProgID of the appropriate OLE object. *end example*] The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleLink) is located in §A.2. *end note*]

18.14.12 row (Row)

This element contains data for an external worksheet row.

Attributes	Description
r (Row)	Row number of the row in the external book containing the cell data referenced. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalRow) is located in §A.2. *end note*]

18.14.13 sheetData (External Sheet Data Set)

This element contains the cached worksheet data associated with a supporting workbook.

[*Note*: For an example, please refer to example at the beginning of this section. *end note*]

Attributes	Description
refreshError (Last Refresh Resulted in Error)	Specifies that the last external data refresh for this sheet did not succeed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
sheetId (Sheet Id)	Index of sheet in the external workbook that is referenced and partially cached in this data set. This is a 1-based index. This attribute is required. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalSheetData) is located in §A.2. *end note*]

18.14.14 sheetDataSet (Cached Worksheet Data)

This element serves as the collection for 1 or more sheetData elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalSheetDataSet) is located in §A.2. *end note*]

18.14.15 sheetName (Sheet Name)

Name of a worksheet in the supporting workbook

Attributes	Description
val (Sheet Name Value)	Name of the sheet. This attribute is required. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalSheetName) is located in §A.2. *end note*]

18.14.16 sheetNames (Supporting Workbook Sheet Names)

This element is the container for all of the worksheet names in a supporting workbook.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalSheetNames) is located in §A.2. *end note*]

18.14.17 val (DDE Link Value)

This element specifies a value associated with a particular DDE item.

[*Example*: Here's an example of how values, value, and val elements are written out in the spreadsheetML for a ddeItem supplied by a DDE server. In this example different cells in the workbook are bound to these specific DDE items:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<ddeLink xmlns:r="…" ddeService="StockSrv" ddeTopic="Prices">
  <ddeItems>
    <ddeItem name="Bread" advise="1">
      <values>
        <value>
          <val>3.5</val>
        </value>
      </values>
    </ddeItem>
    <ddeItem name="Milk" advise="1">
      <values>
        <value>
          <val>5.7400000000000002</val>
        </value>
      </values>
    </ddeItem>
    <ddeItem name="MSFT" advise="1">
      <values>
        <value>
          <val>54.130000000000003</val>
        </value>
      </values>
    </ddeItem>
    <ddeItem name="StdDocumentName" ole="1" advise="1"/>
  </ddeItems>
</ddeLink>
```

end example]

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.14.18 value (Value)

This element contains a value associated with a particular DDE item. This serves as a container for the val element.

Attributes	Description
t (DDE Value Type)	Indicates the DDE value type. The possible values for this attribute are defined by the ST_DdeValueType simple type (§18.18.23).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_DdeValue) is located in §A.2. *end note*]

18.14.19 values (DDE Name Values)

This element defines a collection of values associated with DDE item.

Attributes	Description
cols (Columns)	The number of columns of data that is returned by the DDE server for this DDE item. The default value of this attribute is 1. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rows (Rows)	The number of rows of data that is returned by the DDE server for this DDE item. The default value of this attribute is 1. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DdeValues) is located in §A.2. *end note*]

18.15 Volatile Dependencies

The volatileDependencies part provides a cache of data that supports Real Time Data (RTD) and CUBE functions in the workbook. Both of these types of functions require connectivity to external servers to retrieve their data. For RTD functions, an RTD interface defines how data is provided on the server, and how it is retrieved on the client. Similarly, CUBE functions access data in OLAP cubes via their own function syntax. The volatileDependencies part provides that cache of data and supporting information about these functions and their data servers and connections. This allows the spreadsheet application to work with cached values when recalculating the workbook when the external server is not available.

[*Note*: How users of SpreadsheetML access RTD data depends on the integration the user's spreadsheet application provides for RTD. *end note*]

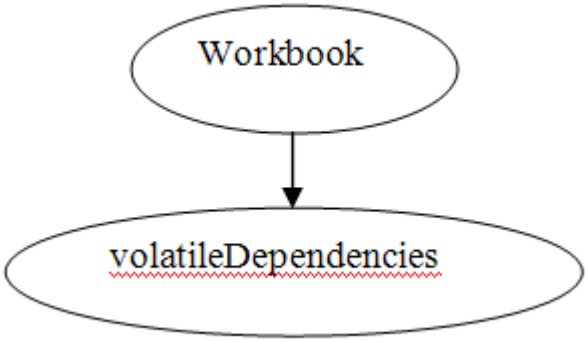
[*Note*: Data connectivity can use a number of different technologies. One example of potential values stored in this attribute can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbaxl11/html/xlobjIRtdServer_HV03085058.asp *end note*]

File Architecture

The workbook holds the relationship to the volatile dependencies part.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



Illustration

[*Example*: The following image shows an example implementation of CUBE and Real Time Data (RTD) functions in a worksheet.

	A
1	=RTD("jrtdx.rtd",,"aaa")
2	
3	=CUBEMEMBER("xlextdat9 Adventure Works DW Adventure Works","[Department].[Departments].[Corporate]")
4	=CUBEVALUE("xlextdat9 Adventure Works DW Adventure Works",A3)
5	=CUBESET("xlextdat9 Adventure Works DW Adventure Works","[Customer].[Customer Geography].[All Customers].[United Kingdom].children","Set")
6	=CUBERANKEDMEMBER("xlextdat9 Adventure Works DW Adventure Works",$A$3,ROW(A1))
7	=CUBESETCOUNT(A3)
8	=CUBEMEMBERPROPERTY("xlextdat9 Adventure Works DW Adventure Works","[Product].[Product].[All Products].[Blade]","Class")
9	=CUBEKPIMEMBER("xlextdat9 Adventure Works DW Adventure Works","Growth in Customer Base",2)
10	

The following example shows the XML that describes the functions in the illustration.

```
<volTypes xmlns="…">
  <volType type="realTimeData">
    <main first="jrtdx.rtd">
      <tp t="s">
        <v>aaa: 4447</v>
        <stp/>
        <stp>aaa</stp>
        <tr r="A1" s="1"/>
      </tp>
    </main>
  </volType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <volType type="olapFunctions">
      <main first="xlextdat9 Adventure Works DW Adventure Works">
        <tp t="e">
          <v>#N/A</v>
          <stp>1</stp>
          <tr r="A6" s="1"/>
          <tr r="A9" s="1"/>
          <tr r="A8" s="1"/>
          <tr r="A5" s="1"/>
          <tr r="A4" s="1"/>
          <tr r="A3" s="1"/>
        </tp>
      </main>
    </volType>
  </volTypes>
```

end example]

While RTD and Cube functions share the cache, there are differences in how the data is interpreted. [*Example*: RTD dependencies, volTypes/volType/main@first specifies the ProgId of the RTD server. Whereas for OLAP dependencies, main@first indicates the connection name. *end example*]

18.15.1 main (Main)

Represents dependency information for all topics within a volatile dependency type that share the same first string or function argument.

Attributes	Description

first (First String)	Specifies the first string of all topics within this main. This string corresponds to the first argument to the RTD or CUBE function.
	
	For RTD functions, this argument represents the progID of the IRTDServer.
	
	[*Example*:
	`<main first="jrtdx.rtd">`
	end example]
	
	For CUBE functions, this argument represents the CUBE connection.
	
	[*Example*:
	`<main first="xlextdat9 Adventure Works DW Adventure Works">`
	end example]
	
	For more information on RTD and CUBE functions in SpreadsheetML, see §18.17 in Formulas.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_VolMain) is located in §A.2. *end note*]

18.15.2 stp (Strings in Subtopic)

Represents all strings in the topic except for the first. An stp is allocated for each additional argument. [*Example*: For the topic {"progid","","foo"}, there would be two STPs: "" and "foo". *end example*]

For Cube functions, value of "1" indicates that all of the related cells with calling cube functions have been refreshed.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

18.15.3 tp (Topic)

Represents dependency information for all topics within a volatile dependency type that share the same first string or argument.

For the RTD function, this collection will contain the remaining parameters of the function, and indicate the last known value and data type of that value.

Attributes	Description

t (Type)	Specifies the type of the cell value. This value corresponds to the type of data returned by the RTD or CUBE function.
	
	[*Example*: In the following RTD example, the value "aaa: 4447" has a string data type.
	
	`<tp t="s">`
	`  <v>aaa: 4447</v>`
	`</tp>`
	
	end example]
	
	For Cube functions, this attribute can be ignored when stp value is "1".
	
	The possible values for this attribute are defined by the ST_VolValueType simple type (§18.18.91).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_VolTopic) is located in §A.2. *end note*]

18.15.4 tr (References)

Represents the reference to a cell that depends on this topic. Each topic can have one or more cells dependencies.

For CUBE functions, each <tr> element contains a cell whose cube function call dependent on the connection in main@first.

Attributes	Description
r (Reference)	Specifies a reference to the cell location. The location is scoped to the sheet specified in s. [*Example*: `<tr r="A6" s="1"/>` *end example*] The possible values for this attribute are defined by the ST_CellRef simple type (§18.18.7).
s (Sheet Id)	Specifies the sheet index. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_VolTopicRef) is located in §A.2. *end note*]

18.15.5 volType (Volatile Dependency Type)

Represents dependency information for a specific type of external data server. There is no limit on the number of external dependencies that can exist for a workbook in SpreadsheetML.

Attributes	Description
type (Type)	Specifies the type of the external dependency. [*Example*: `<volType type="olapFunctions">` *end example*] The possible values for this attribute are defined by the ST_VolDepType simple type (§18.18.90).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_VolType) is located in §A.2. *end note*]

18.15.6 volTypes (Volatile Dependency Types)

Represents the collection of external dependencies for a workbook. This element defines the structure of the volatilateDependencies part. There can only be one volatileDependencies part for each workbook. However, the part can contain one or more dependency types.

The volatileDependencies part stores the following information for Real Time Data (RTD) and CUBE functions:

- Cached values
- Parameters used
- Connection and Server names

[*Example*: *Outline of XML Structure*

```
<volTypes xmlns="…">
  <volType type="realTimeData">
  </volType>
  <volType type="olapFunctions">
  </volType>
</volTypes>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_VolTypes) is located in §A.2. *end note*]

18.16 Custom XML Mappings

Custom XML Mappings enable binding of arbitrary XML data structures and arbitrary XML schema definitions to the workbook. Once a DataBinding has been established, then various XML nodes can be mapped to table columns, ranges of cells, or even single cells (for non-repeating attributes and elements). Once an XML Mapping is fully defined, the application is able to import and export XML instance structures according to the schema definition. ISO/IEC 29500 does not require any particular XML schema language.

[*Note:* Some examples of XML schema languages that might be used to implement Custom XML Mappings include:

- W3C XML Schema - http://www.w3.org/XML/Schema
- RELAX NG – ISO/IEC 19757-2
- Schematron – ISO/IEC 19757-3
- NVDL – ISO/IEC 19757-4

end note]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

While the original schema or XML definition can reside on disk or at some file location outside the workbook, a copy of the schema is stored in the workbook.

Every time an XML instance or schema is added to the workbook, a new map object is created which ties together the schemas and where the various elements are mapped in the workbook.

[*Example*:

```
<MapInfo SelectionNamespaces="">
  <Schema ID="Schema1">
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element nillable="true" name="Root">
        <xsd:complexType>
          <xsd:sequence minOccurs="0">
            <xsd:element minOccurs="0" nillable="true" name="EmployeeInfo"
              form="unqualified">
              <xsd:complexType>
                <xsd:sequence minOccurs="0">
                  <xsd:element minOccurs="0" nillable="true" type="xsd:string"
                    name="Name" form="unqualified"></xsd:element>
                  <xsd:element minOccurs="0" nillable="true" type="xsd:date"
                    name="Date" form="unqualified"></xsd:element>
                  <xsd:element minOccurs="0" nillable="true" type="xsd:integer"
                    name="Code" form="unqualified"></xsd:element>
                </xsd:sequence>
              </xsd:complexType></xsd:element>

              <xsd:element minOccurs="0" maxOccurs="unbounded" nillable="true"
                name="ExpenseItem" form="unqualified">
                <xsd:complexType>
                  <xsd:sequence minOccurs="0">
                    <xsd:element minOccurs="0" nillable="true" type="xsd:date"
                      name="Date" form="unqualified"></xsd:element>
                    <xsd:element minOccurs="0" nillable="true" type="xsd:string"
                      name="Description" form="unqualified"></xsd:element>
                    <xsd:element minOccurs="0" nillable="true" type="xsd:double"
                      name="Amount" form="unqualified"></xsd:element>
                  </xsd:sequence>
                </xsd:complexType></xsd:element>
            </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:attribute name="Currency" form="unqualified"
              type="xsd:string"></xsd:attribute>
            <xsd:attribute name="Approved" form="unqualified"
              type="xsd:string"></xsd:attribute>
          </xsd:complexType>
        </xsd:element>
      </xsd:schema>
    </Schema>
    <Map ID="1" Name="Root_Map" RootElement="Root" SchemaID="Schema1"
      ShowImportExportValidationErrors="false" AutoFit="true" Append="false"
      PreserveSortAFLayout="true" PreserveFormat="true">
      <DataBinding ConnectionID="1" FileBinding="true" DataBindingLoadMode="1"/>
    </Map>
  </MapInfo>
```

end example]

For XML mapped into a SpreadsheetML Table there will also be additional information in the SpreadsheetML file which refers back to the XML Map and XPath of the element or attribute mapped. This information is stored in a xmlColumnPr element, under the tableColumn node.

[*Example*:

```
  <table xmlns="…" id="1" name="Table1" displayName="Table1" ref="A1:H11"
    tableType="xml" totalsRowShown="0" connectionId="1">
    <tableColumns count="5">
      <tableColumn id="1" uniqueName="Name" name="Name">
        <xmlColumnPr mapId="1" xpath="/Root/EmployeeInfo/Name"
          xmlDataType="string"/>
      </tableColumn>
      <tableColumn id="2" uniqueName="Date" name="Date">
        <xmlColumnPr mapId="1" xpath="/Root/EmployeeInfo/Date"
          xmlDataType="date"/>
      </tableColumn>
      <tableColumn id="3" uniqueName="Code" name="Code">
        <xmlColumnPr mapId="1" xpath="/Root/EmployeeInfo/Code"
          xmlDataType="integer"/>
      </tableColumn>
      <tableColumn id="4" uniqueName="Description" name="Description">
        <xmlColumnPr mapId="1" xpath="/Root/ExpenseItem/Description"
          xmlDataType="string"/>
      </tableColumn>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <tableColumn id="5" uniqueName="Amount" name="Amount">
        <xmlColumnPr mapId="1" xpath="/Root/ExpenseItem/Amount"
          xmlDataType="double"/>
      </tableColumn>
    </tableColumns>
    <tableStyleInfo name="TableStyleMedium9" showFirstColumn="0"
      showLastColumn="0" showRowStripes="1" showColumnStripes="0"/>
  </table>
```

end example]

For XML mapped into a single SpreadsheetML cell there will also be additional information in the TableSingleCells part which refers back to the XML Map and XPath of the element or attribute mapped. This information is stored in the xmlPr element under the xmlCellPr node.

[*Example*:

```
  <singleXmlCells xmlns="…">
    <singleXmlCell id="2" name="Table2" displayName="Table2" r="D19"
      connectionId="1">
      <xmlCellPr id="1" uniqueName="Currency">
        <xmlPr mapId="1" xpath="/Root/@Currency" xmlDataType="string"/>
      </xmlCellPr>
    </singleXmlCell>
    <singleXmlCell id="3" name="Table3" displayName="Table3" r="D20"
      connectionId="1">
      <xmlCellPr id="1" uniqueName="Approved">
        <xmlPr mapId="1" xpath="/Root/@Approved" xmlDataType="string"/>
      </xmlCellPr>
    </singleXmlCell
    <singleXmlCell id="4" name="Table4" displayName="Table4" r="D18"
      connectionId="1">
      <xmlCellPr id="1" uniqueName="Name">
        <xmlPr mapId="1" xpath="/Root/EmployeeInfo/Name" xmlDataType="string"/>
      </xmlCellPr>
    </singleXmlCell>
  </singleXmlCells>
```

end example]

18.16.1 DataBinding (XML Mapping)

This element contains properties which specify how the XML mapping should work.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<DataBinding ConnectionID="1" FileBinding="true" DataBindingLoadMode="1"/>
```

end example]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
ConnectionID (Reference to Connection ID)	Specifies the Connection ID to the external connection in the External Data Connections part. Required if FileBinding is true. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
DataBindingLoadMode (XML Data Loading Behavior)	Specifies the mode for loading XML data related to this DataBinding. Supported values are as follows: 0 - None 1 - Normal 2 - Delay Load 3 - Asynchronous 4 - Object Model The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
DataBindingName (Unique Identifer)	Specifies the data binding name. These shall be unique for each DataBinding. [*Example*: `<DataBinding DataBindingName="Binding1" FileBinding="true" FileBindingName="Binding1" DataBindingLoadMode="1"/>` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
FileBinding (Binding to External File)	Specifies whether the data should be retrieved directly from an XML file. The path to the file is in the corresponding connection element The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
FileBindingName (File Binding Name)	Specifies the file binding name. These shall be unique for each DataBinding. [*Example*: `<DataBinding DataBindingName="Binding1" FileBinding="true" FileBindingName="Binding1" DataBindingLoadMode="1"/>` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataBinding) is located in §A.2. *end note*]

18.16.2 Map (XML Mapping Properties)

This element contains all of the properties related to the XML map, and the behaviors expected during data refresh operations.

[*Example*:

```
<Map ID="1" Name="Root_Map" RootElement="Root" SchemaID="Schema1"
  ShowImportExportValidationErrors="false" AutoFit="true" Append="false"
  PreserveSortAFLayout="true" PreserveFormat="true">
  <DataBinding ConnectionID="1" FileBinding="true" DataBindingLoadMode="1"/>
</Map>
```

end example]

Attributes	Description
Append (Append Data to Table)	Specifies whether XML data should overwrite or be appended to the end of the table or range of mapped cells when data is refreshed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
AutoFit (AutoFit Table on Refresh)	Specifies whether columns should be resized to fit the XML data after a data refresh operation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ID (XML Mapping ID)	Specifies the ID of the XML map. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
Name (XML Mapping Name)	Specifies the name of the XML map. The possible values for this attribute are defined by the W3C XML Schema string datatype.
PreserveFormat (Preserve Cell	Specifies whether cell number formatting in the sheet should be preserved during data refresh operations, or whether the number formatting defined by the XML data type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Formatting)	should be used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
PreserveSortAFLayout (Preserve AutoFilter State)	Specifies whether to keep the filter state of the Table or cell range intact during a data refresh. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
RootElement (Root Element Name)	Specifies the names of the root XML element. The possible values for this attribute are defined by the W3C XML Schema string datatype.
SchemaID (Schema Name)	Specifies the unique name of the schema used for the mapping. The possible values for this attribute are defined by the W3C XML Schema string datatype.
ShowImportExportValidationErrors (Show Validation Errors)	Specifies whether XML schema validation errors should be displayed during data refresh or data export. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Map) is located in §A.2. *end note*]

18.16.3 MapInfo (XML Mapping)

This element acts as the container for all of the XML schemas and maps attached to the SpreadsheetML document.

Attributes	Description

SelectionNamespaces (Prefix Mappings for XPath Expressions)	Specifies namespaces for use in XPath expressions when it is necessary to define new namespaces externally. Namespaces are defined in the XML style, as a space-separated list of namespace declaration attributes
	
	[*Example*: The following example contains elements that belong to "a" and "b", in addition to elements that do not belong to any namespace.
	
	`<?xml version="1.0"?>`
	`<root>`
	`<branch>branch</branch>`
	`<a:root xmlns:a="http://myserver.com">`
	`<a:branch>a-branch</a:branch>`
	`<b:branch xmlns:b="http://yourserver.com">`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	b-branch</b:branch> </a:root> </root> *end example*] This is used when writing Xpath expressions at runtime against the XML instance structures, because the Xpath expressions use namespace prefixes instead of the fully spelled out namespace. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MapInfo) is located in §A.2. *end note*]

18.16.4 Schema (XML Schema)

This element contains the XML tree for an attached schema.

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
ID (Schema ID)	Specifies the unique name or ID for this attached schema. [*Example*: ID = "Schema1" *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
Namespace (Schema Root Namespace)	Specifies the namespace used by the schema. [*Example*: <MapInfo SelectionNamespaces="…"> <Schema ID="Schema1" Namespace="…"> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
SchemaLanguage (Schema Language)	Specifies the media type of the schema language. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<Schema … SchemaLanguage="application/relax-ng-compact-syntax"/> *end example*] The possible values for this attribute are defined by the W3C XML Schema token datatype.
SchemaRef (Schema Reference)	The schemaRef attribute is used in the specific case where the schema definition happens to include another schema file that contributes to the same namespace. The value of this attribute is the relative path to a "root" schema file on disk which in turn references the other schema files contributing type definitions to the same namespace. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Schema) is located in §A.2. *end note*]

18.17 Formulas

18.17.1 Introduction

A SpreadsheetML *formula* is the syntactic representation of a series of calculations that is parsed or interpreted by the spreadsheet application into a function that calculates a value or array of values based upon zero-to-many inputs.

A formula is an expression that can contain the following: constants, operators, cell references, calls to functions, and names.

[*Example*: Consider the formula `PI()*(A2^2)`. In this case,

- `PI()` results in a call to the function `PI`, which returns the value of π.
- The cell reference A2 returns the value in that cell.
- 2 is a numeric constant.
- The caret (^) operator raises its left operand to the power of its right operand.
- The parentheses, `(` and `)`, are used for grouping.
- The asterisk (*) operator performs multiplication of its two operands.

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.2 Syntax

The syntax rules in this subclause follow the system shown in ISO/IEC 14977: literal text is surrounded by double-quotes (or by apostrophes); the left-square-bracket and right-square-bracket designate the start and end of an option; the left-curly-bracket and right-curly-bracket designate the start and end of an sequence of one-or-more items; the vertical-line indicates an alternative; and each rule ends with a semicolon. Whenever hyphen is used as the exception-symbol (as per ISO/IEC 14977), it is surrounded by white space, and further clarified by a comment.

The syntax rules below are modified by the context-sensitive rules stated in other subclauses of §18.17.*, as indicated by an EBNF comment of the form "(* see semantic rules at 18.17… *)". [*Note*: Thus, in order to produce an automated verifier, the context-sensitive rules in subclauses 18.17.* must all be considered. The context-free syntax rules shown in this subclause provide an overview to assist comprehension by the reader, but do not represent the entirety of the context-sensitive rules. *end note*]

When used in narrative, production names are set in an italic style, as in *array-constant*, *expression*, and *function-name*.

The syntax of a formula is as follows:

```
formula=
  expression  ;
expression=
  "(",  expression,  ")"  |
  constant  |
  prefix-operator,  expression  |
  expression,  infix-operator,  expression  |
  expression,  postfix-operator  |
  cell-reference  |
  function-call  |
  name  ;
```

where *expression* is an arbitrarily complex expression involving constants (§18.17.2.1), operators (§18.17.2.2), cell references (§18.17.2.3), calls to functions (§18.17.2.4), and names (§18.17.2.5).

A *token* is the minimal lexical element of a formula. The categories of tokens are: constants (except for *array-constant*), operators, cell references, function names, names, and punctuators. The *punctuators* are:

- Left parenthesis (`(`) and right parenthesis (`)`) used for expression grouping and in a function call.
- Comma (`,`) used in a function call and an *array-constant*.
- Left brace (`{`), right brace (`}`), and semicolon (`;`) used in an *array-constant*.

In a formula, an arbitrary number of space characters (`U+0020`) can precede the first token or follow the final token. An arbitrary number of space characters can separate two adjacent tokens, except that no space characters shall separate a *function-name* from the left parenthesis (`(`) that follows it. Such space characters

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

have no effect on the semantics of a formula; however, such spaces shall be distinguished from the space operator (§18.17.2.2).

All arithmetic terms in an *expression* are numbers that can be represented by the workbook's value space. For any given workbook, the default value space is IEC 60559's double precision. However, an implementation is permitted to override that default value space by use of the characteristics markup (§22.7). [*Example*: In the expression `1/3`, although the operands appear to be integers, they are, in fact, floating-point numbers, and the result is `0.33…`, not `0`, as would result from integer division. *end example*]

As ranges of data are fundamental to spreadsheet calculations, many SpreadsheetML functions are able to take arrays as inputs and to return arrays as outputs. Based on whether the formula is an *array formula* or not, the way in which input ranges are interpreted and output values are understood to relate to cells sharing the formula can mean one of two things.

For an array formula: All range arguments are interpreted to be their full range. If the result of the formula is an array, the values of the array are meant to be returned across all of the cells in the sheet sharing the formula. (When the size of the range for an array formula exceeds in either dimension the size of the returned array, the excess cells take on a value of `#N/A.`)

For a normal (non-array) formula:

- Implicit intersection (§18.17.2.2) is performed on all arguments to functions except for those that allow a range.
- If the formula results in an array, only the first value from the array is returned to the cell.

Implicit intersection is determined as follows: When a range is passed to a function which expects only a single cell, a test is made to discover whether the calling cell intersects that range at any point horizontally or vertically. If it does, the cell at the point of intersection is passed to the function. [*Example*: The formula `ABS(B1:B3)` is entered into A2. Because the ABS function does not expect a range, implicit intersection is performed. A2 intersects `B1:B3` horizontally on row 2, and so the value in B2 is passed into the function. *end example*]

The list of function arguments that allow a range is as follows:

- AND - all arguments
- AREAS - *reference* argument
- AVEDEV - all arguments
- AVERAGE - all arguments
- AVERAGEA - all arguments
- AVERAGEIF - all arguments except for *criteria*
- AVERAGEIFS - all arguments except for *criteria1*, *criteria2*, etc.
- CELL - *reference* argument
- CHITEST - all arguments
- COLUMN - all arguments

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- COLUMNS - all arguments
- CORREL - all arguments
- COUNT - all arguments
- COUNTA - all arguments
- COUNTBLANK - all arguments
- COUNTIF - all arguments except *criteria*
- COUNTIFS - all arguments except for *criteria1*, *criteria2*, etc.
- COVAR - all arguments
- DAVERAGE - all arguments
- DCOUNT - all arguments
- DCOUNTA - all arguments
- DEVSQ - all arguments
- DGET - all arguments
- DMAX - all arguments
- DMIN - all arguments
- DPRODUCT - all arguments
- DSTDEV - all arguments
- DSTDEVP - all arguments
- DSUM - all arguments
- DVAR - all arguments
- DVARP - all arguments
- FORECAST - all arguments except for *x*
- FREQUENCY - all arguments
- FTEST - all arguments
- FVSCHEDULE - *schedule* argument
- GCD - all arguments
- GEOMEAN - all arguments
- GROWTH - all arguments
- HARMEAN - all arguments
- HLOOKUP - *table_array* argument
- IMPRODUCT - all arguments
- IMSUM - all arguments
- INDEX - *array* or *reference* argument
- INTERCEPT - all arguments
- LARGE - *array* argument
- LCM - all arguments
- LINEST - *known-xs* and *known-ys* arguments
- LOGEST - *known-xs* and *known-ys* arguments
- LOOKUP - all arguments except *lookup_value*
- MATCH - *lookup_array* argument

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- MAX - all arguments
- MAXA - all arguments
- MDETERM – *array* argument
- MEDIAN - all arguments
- MIN - all arguments
- MINA - all arguments
- MINVERSE - all arguments
- MIRR - *values* argument
- MMULT - all arguments
- MODE - all arguments
- MULTINOMIAL - all arguments
- NETWORKDAYS - *holidays* argument
- NETWORKDAYS.INTL - *holidays* argument
- NPV - all arguments except *rate*
- OFFSET - *reference* argument
- OR - all arguments
- PEARSON - all arguments
- PERCENTILE - *array* argument
- PERCENTRANK - *array* argument
- PROB - *x_range* and *prob_range* arguments
- PRODUCT - all arguments
- QUARTILE – *array* argument
- RANK - *ref* argument
- ROW – *reference* argument
- ROWS – *array* argument
- RSQ - all arguments
- SKEW - all arguments
- SLOPE - all arguments
- SMALL - *array* argument
- STDEV - all arguments
- STDEVA - all arguments
- STDEVP - all arguments
- STDEVPA - all arguments
- STEYX - all arguments
- SUBTOTAL - all arguments except *function_num*
- SUM - all arguments
- SUMIF - all arguments except *criteria*
- SUMIFS - all arguments except *criteria1*, *criteria2*, etc.
- SUMPRODUCT - all arguments
- SUMSQ - all arguments

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- SUMX2MY2 - all arguments
- SUMX2PY2 - all arguments
- SUMXMY2 - all arguments
- TRANSPOSE – all arguments
- TREND – all arguments except *const-flag*
- TRIMMEAN – *array* argument
- TRUNC – *x* argument
- TTEST – *array-1* and *array-2* arguments
- TYPE – all arguments
- VALUE – all arguments
- VAR – all arguments
- VARA – all arguments
- VARP – all arguments
- VARPA – all arguments
- VLOOKUP – *table-array* argument
- WORKDAY – *holidays* argument
- WORKDAY.INTL – *holidays* argument
- XIRR – all arguments except *guess*
- XNPV – all arguments except *rate*
- ZTEST – *array* argument

[*Example*: Here are some formulas taking array constants and ranges:

- `(B2:B4*C2:C4)+10.5` performs three calculations: `(B2*C2)+10.5`, `(B3*C3)+10.5`, and `(B4*C4)+10.5`.
- `SQRT({1,2,3,4})` returns 1 when entered normally.
- `SQRT({1,2,3,4})` returns 1 when used in an array formula in a single cell, but if the array formula spans four or more contiguous cells in a row, it will return 1, 1.41, 1.73, and 2 in the first four cells, respectively, and `#N/A` in any additional cells in the horizontal range spanned by the array formula. (For display purposes, the values returned have been truncated to two decimal places.)
- `SUM(SQRT({1,2,3,4}))` returns 6.14 when entered normally; when used in an array formula, the intermediate calculation (in this instance performed by the SQRT function) is performed upon each element of the array.

With `A1:A4` holding the values 1, 2, 3, and 4, respectively:

- `SQRT(A1:A4)` entered normally will do implicit intersection if it is in any of the rows 1–4, and return the SQRT of the number in the same row.
- `SQRT(A1:A4)` returns 1 when used in an array formula in a single cell, since it does not do implicit intersection in this case. If it is used in an array formula spanning multiple contiguous cells in a column, it will return 1, 1.41, 1.73, 2, `#N/A`, …, respectively, in the cells in its vertical output range. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.2.1 Constants

A *constant* represents a fixed value that can be used in the calculation of a formula. A constant has the following form:

```
constant=
  error-constant  |
  logical-constant  |
  numerical-constant  |
  string-constant  |
  array-constant  ;
error-constant=
  "#DIV/0!"  |  "#N/A"  |  "#NAME?"  |  "#NULL!"  |
  "#NUM!"  |  "#REF!"  | "#VALUE!" |  "#GETTING_DATA"  ;
logical-constant=
  "FALSE"  |  "TRUE"  ;
numerical-constant=
  whole-number-part,  [full-stop],  [exponent-part]  |
  full-stop,  fractional-part,  [exponent-part]  |
  whole-number-part,  full-stop,  fractional-part  [exponent-part]  ;
full-stop=
  "."    ;  (* also known as “period” *)
whole-number-part=
  digit-sequence  ;
fractional-part=
  digit-sequence  ;
exponent-part=
  "e" [ sign ] digit-sequence  |
  "E" [ sign ] digit-sequence  ;
sign=
  "+"  |
  "-"  ;
digit-sequence=
  decimal-digit,  {decimal-digit}  ;
decimal-digit=
  "0"  |  "1"  |  "2”  |  "3"  |  "4"  |
  "5”  |  "6"  |  "7"  |  "8”  |  "9"  ;
string-constant=
  double-quote, [string-chars], double-quote  ;
double-quote=
  '""'  ; (* one double-quote character*)
string-chars=
  string-char,  {string-char}  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
string-char=
  '""'  |   (* consecutive double-quotes, with no space between them *)
  character  -  double-quote  ;   (* any character except double-quote *)
character=
  as defined by the production Char in the XML 1.0 specification, §2.2.
```

[*Example*: =$A$1/3 divides the value in A1 by the constant value three; =$A$1&"a" appends the constant string "a" to the string representation of the contents of cell A1. *end example*]

To include a double-quote character (") in *string-chars*, precede it with another double-quote character. [*Example*: `"ab""cd"` contains the characters `ab"cd`, and `"""abcd"""` contains the characters `"abcd"`. *end example*]

An *array constant* is a list of one or more constants organized in one or two dimensions, and delimited by braces. An array constant has the following form:

```
array-constant=
  "{",  constant-list-rows,  "}"  ;
constant-list-rows=
  constant-list-row,  {  semicolon,  constant-list-row  }  ;
semicolon=
  ";"    ;
constant-list-row=
  constant,  {  comma,  constant  }  ;
```

An *array-constant* shall not contain

- An *array-constant*.
- Columns or rows of unequal length.

Any *numerical-constant* in an *array-constant* can be preceded immediately by a *prefix-operator*.

The *constants* in an *array-constant* can have different types.

[G*uidance* An implementation is encouraged to not unnecessarily limit the number of rows and columns in an *array-constant*. *end guidance*]

[*Example*: `{1,3.5,TRUE,"Hello"}` is a 1x4 array of constants.

To represent the values 10, 20, 30, and 40, as a 1x4 array, use `{10,20,30,40}`.

To represent the values 10, 20, 30, and 40 in the first row, and 50, 60, 70, and 80 in the second row, use the following 2x4 array constant: `{10,20,30,40;50,60,70,80}`. *end example*]

error-constant is described in §18.17.3.

Each *constant* has a corresponding type (§18.17.2.6), as follows:

IECNORM.COM: Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Constant Form	Type
array-constant	array
error-constant	error
logical-constant	logical
numerical-constant	number
string-constant	text

In the context of cell formulas and values in SpreadsheetML, the following definition of precision shall apply:

By default, default representation of precision shall be as defined by the XML schema double type: http://www.w3.org/TR/xmlschema-2/#double. The default is therefore 53-bits of mantissa precision.

An application that uses XML schema double can optionally state the precision in the Additional Characteristics part by writing out the number of bits in the mantissa and exponent.

A compliant consumer shall parse numbers of arbitrary precision without error.

18.17.2.2 Operators

An *operator* is a symbol that specifies the type of operation to perform on one or more operands. There are arithmetic, comparison, text, and reference operators.

```
operator=
  ":"  |  comma  |  space  |  "^"  |  "*"  |  "/"  |  "+"  |  "-"
  |  "&"  |  "="  |  "<>"  |  "<"  |  "<="  |  ">"  |  ">="  |  “%”  ;
infix-operator=
  ":"  |  comma  |  space  |  "^"  |  "*"  |  "/"  |  "+"  |  "-"  |  "&"
  |  "="  |  "<>"  |  "<"  |  "<="  |  ">"  |  ">="  ;
postfix-operator=
  "%"  ;
prefix-operator=
  "-"  ;
```

The *operator*s permitted in *expression* are:

Operators			
Family	Operator	Description	Precedence
Reference operators	:	Binary range operator, which takes two cell reference (§18.17.2.3) operands, and results in one reference to the cells inclusive of, and between, those references. [*Example*: `SUM(B5:C15)`, which references 11 cells. *end example*]	highest
	,	Binary union operator, which takes two cell reference (§18.17.2.3) operands, and results	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Operators			
		in one reference to all those, possibly non-contiguous, cells. [*Example*: `SUM((B5:B15,D5:D15))`, which references 22 cells, 11 from column B, and 11 from column D. The grouping parentheses are necessary to indicate that the comma is an operator rather than a punctuator separating two arguments. *end example*]	
	space	Binary intersection operator, which takes two cell reference (§18.17.2.3) operands, and results in one reference to those, possibly non-contiguous, cells that are common. If the intersection is empty, the result is `#NULL!`. [*Example*: `((B1:C1) (C1:D1))` results in a reference to `C1`, while `((B1:D1) (B1,D1))` results in a single reference to `B1` and `D1`. *end example*]	
Arithmetic operators	-	Unary minus	
	%	Percentage (unary postfix), which divides its operand by 100. [*Example*: `10.5%`, which results in `0.105`. *end example*]	
	^	Exponentiation	
	*	Multiplication	
	/	Division	
	+	Addition	
	-	Subtraction	
Text operator	&	Text concatenation (Each of the two operands is converted to text, if necessary, before concatenation.)	
Comparison operators	=	Equal-to	lowest
	<>	Not-equal-to	
	<	Less-than	
	<=	Less-than or equal-to	
	>	Greater-than	
	>=	Greater-than-or-equal-to	

expression can contain grouping parentheses to document the default precedence or to override it.

*operator*s in *expression* having the same precedence associate left-to-right.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Given that cell E38 contains the value 4, and cell F38 contains the value 2, the formula

```
((-1+E38^2)*3-F38)/2
```

produces the result 21.5. *end example*]

The comparison operators yield TRUE for true and FALSE for false. An expression with value 0 tests logically false while one with any non-zero numeric value tests true.

For any given operator in an expression, if only one operand is an error value, the result is that error value. If more than one operand has an error value and those error values are the same, the result is that error value. If more than one operand has an error value and those error values are not all the same, as to which of those error values is used as the result is unspecified.

It the semantics of an operator having a given operand are not specified by ISO/IEC 29500, the result is #VALUE!. [*Example*: "abc"+1 results in #VALUE!, and "abc"/0 results in #VALUE! rather than #DIV/0!. *end example*]

18.17.2.3 Cell References

Each set of horizontal cells in a worksheet is a *row*, and each set of vertical cells is a *column*. A cell's row and column combination designates the location of that cell. [*Guidance*: An implementation is encouraged to not unnecessarily limit the number of rows and columns in a worksheet. *end guidance*]

A *cell reference* designates one or more cells on the same worksheet. Using references, one can:

- Use data contained in different parts of the same worksheet in a single formula.
- Use the value from a single cell in several formulas.
- Refer to cells on other sheets in the same workbook, and even to other workbooks. (References to cells in other workbooks are called *links*.)

A cell reference has the following form:

```
cell-reference=
  name       |
  [work-sheet-prefix]  A1-reference      |
  [work-sheet-prefix]  A1-reference,  ":",  A1-reference      |
  [work-sheet-prefix]  R1C1-reference    |
  [work-sheet-prefix]  R1C1-reference,  ":",  R1C1-reference    ;
work-sheet-prefix=
  work-sheet-prefix-special  |
  sheet-name,  "!"  |
  sheet-name,  ":",  sheet-name,  "!"  |
  "[",  workbook-name,  "]",  sheet-name,  ":",  sheet-name,  "!"   ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
work-sheet-prefix-special=
  apostrophe,  sheet-name-special,  apostrophe, "!"  |
  apostrophe,  sheet-name-special,  ":",
                      sheet-name-special,  apostrophe,  "!"  |
  apostrophe,  "[",  workbook-name-special,  "]",
                       sheet-name-special,  apostrophe,  "!"  |
  apostrophe,  "[",  workbook-name-special,  "]",
                       sheet-name-special,  ":",  sheet-name-special,
                       apostrophe,  "!"   ;
workbook-name=
  book-name-characters  ;
book-name-characters=
  book-name-character,  {book-name-character}  ;
book-name-character=
  character  -  (operator  |  apostrophe  |  "["  |  "]"  |  "?")  ;
      (* any character except operator or ', [, ], or ? *)
apostrophe=
  "'"  ;   (* one apostrophe character *)
sheet-name=
  sheet-name-characters   ;
sheet-name-character=
  character  -  (operator  |  apostrophe  |  "["  |  "]"  |  "\"  |  "?")  ;
             (* any character except operator or ', [, ], \, or ? *)
sheet-name-characters=
  sheet-name-character,  {sheet-name-character}  ;
workbook-name-special=
  book-name-start-character-special,
    [ book-name-characters-special ];
book-name-start-character-special=
  character  -  (apostrophe  |  "*"  |  "["  |  "]"  |  ":"  |  "?")  ;
      (* any character, including operator, except ', *, [, ], :, or ? *)
book-name-characters-special=
  book-name-character-special,  {book-name-character-special}  ;
book-name-character-special=
  apostrophe, apostrophe  |
  character  -  (apostrophe  |  "*"  |  "["  |  "]"  |  ":"  |  "?")  ;
        (* any character, including operator, except ', *, [, ], :, or ? *)
sheet-name-special=
  sheet-name-start-character-special,
    [ [sheet-name-characters-special], sheet-name-end-character-special]  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
sheet-name-start-character-special=
  character  -  (apostrophe  |  "*"  |  "["  |  "]"  |  "\"  |  ":"  |  "/"
     |  "?")  ;
     (* any character, including operator, except ', *, [, ], \, :, /, or ? *)
sheet-name-end-character-special=
  sheet-name-start-character-special  ;
sheet-name-characters-special=
  sheet-name-character-special,  {sheet-name-character-special}  ;
sheet-name-character-special=
  apostrophe, apostrophe  |
  character  -  (apostrophe  |  "*"  |  "["  |  "]"  |  "\"  |  ":"  |  "/"
     |  "?")  ;
     (* any character, including operator, except ‘ *, [, ], \, :, /, or ? *)
```

A *relative cell reference* is based on the relative position of the cell that contains the formula and the cell to which the reference refers. If the position of the cell that contains the formula changes, the reference is changed along with it.

An *absolute cell reference* always refers to the absolute location of a cell. If the position of the cell that contains the formula changes, the absolute reference remains the same.

A *mixed cell reference* has either an absolute column and relative row, or an absolute row and relative column.

A *link* or *external reference* to a workbook is a reference that specifies the location of the workbook, including file or network path, book name, sheet name, and cell reference. Instead of writing the full file path or network location and workbook name directly in the f (formula) element, in order to make all external references more accessible, the workbook name shall be written in a Relationship part according to the Relationships semantic:

- Type shall be set to http://purl.oclc.org/ooxml/officeDocument/relationships/externalLinkPath,
- Target shall specify the full file path and file name, and
- TargetMode shall be set to `External`.

Additionally, in order to support the possibility that the external workbook is offline or otherwise inaccessible, a cache of the relevant sheet values in the referenced external workbook shall be stored within the referencing workbook according to §18.14.

This Supplementary Workbook Data part shall be the source part of the relationship that points to the external workbook as a target resource, and the Id of this relationship shall be referenced within the markup of the Supplementary Workbook Data part, using the externalBook element (§18.14.7).

The Supplementary Workbook Data part shall also be the target of a relationship whose source is the Workbook part (§18.2). The markup within the Workbook part shall reference this relationship Id using the externalReference element (§18.2.8).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Finally, a 1-based index referencing an externalReference within the collection externalReferences shall be written inline within the formula expression containing the reference to the external workbook. The index in this context shall be enclosed within square brackets, i.e.; left square bracket ([), followed by the index, followed by a right square bracket (]).

In this way, external resource files can more easily be accessed and updated.

[*Example*:

Consider the formula `=SUM('C:\[Source.xlsx]Sheet1'!$A$1:$A$3)`

This formula is expressed in the f element (formula) like this:

```
<f>SUM([1]Sheet1!$A$1:$A$3)</f>
```

The external reference to another workbook in this case is tokenized to `[1]`. The value inside the brackets is a 1-based index to the externalReferences collection in the workbook part, referencing a specific externalReference element.

The corresponding content of externalReferences in the workbook part is:

```
<externalReferences>
    <externalReference r:id="rId4"/>
</externalReferences>
```

The workbook part's externalReferences collection indicates that there is an external workbook reference in this workbook. The Supplementary Workbook Data cache, also stored in this workbook, can be found by following the relationship from the workbook whose id value is `rId4`.

That particular relationship (rId4) is expressed as:

```
<Relationship Id="rId4" Type=
"http://purl.oclc.org/ooxml/officeDocument/relationships/externalLinkPath"
Target="externalLinks/externalLink1.xml"/>
```

The above relationship indicates that the formula is supported by the Supplementary Workbook Data cache located at externalLinks/externalLink1.xml in the package.

The corresponding content in externalLink1.xml follows the markup specified in §18.14, Supplementary Workbook Data. The externalBook element in that markup indicates the id of the relationship that points from the source part externalLink1.xml to the location of the actual external workbook:

```
<externalBook xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
r:id="rId1">
```

That relationship (rId1) is shown here:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<Relationship Id="rId1" Type=
"http://purl.oclc.org/ooxml/officeDocument/relationships/externalLinkPath"
Target="file:///C:/Source.xlsx" TargetMode="External"/>
```

This relationship indicates that the external workbook that the formula references resides on a local drive, at c:\source.xlsx. *end example*]

It is possible to process the same cell or set of cells on multiple worksheets within a workbook, using a *3-D reference*. A reference of this type is made up of the cell reference, preceded by a range of worksheet names, and an exclamation mark character (`!`), in that order. A 3-D reference can be used to refer to cells on other sheets, to defined names, and to create formulas by using the following functions: AVERAGE, AVERAGEA, COUNT, COUNTA, MAX, MAXA, MIN, MINA, PRODUCT, STDEV, STDEVA, STDEVP, STDEVPA, SUM, VAR, VARA, VARP, and VARPA.

3-D references shall not be used in array formulas.

By default, a cell reference is understood to refer to one or more cells in the current worksheet. However, a cell reference can be preceded by its parent worksheet name and an exclamation mark (`!`), in that order. This allows cells in one worksheet to be referenced in another worksheet of the same workbook. [*Example*: The cell reference `MonthlyTotals!D1:D12` might be used from within a sibling (or the same) worksheet of `MonthlyTotals` to refer to those 12 cells. *end example*]

An *area* is a set of rectangular-shaped contiguous cells. An area can be a single cell. [*Example*: `A5` and `B6:C10` each designate one area, and `D3:D5,E12:F15` designates two areas (the comma (`,`) being the union operator). *end example*] [*Note*: The number of areas designated by a cell reference can be obtained by calling the function AREAS (§18.17.7.10). *end note*]

There are two cell reference styles: A1 (§18.17.2.3.1) and R1C1 (§18.17.2.3.2).

18.17.2.3.1 A1-Style Cell References

A cell reference using the A1 reference style has the following form:

```
A1-reference=
  A1-column, ":", A1-column    |
  A1-row,  ":", A1-row    |
  A1-column,  A1-row  ;
A1-column=
  A1-relative-column  |
  A1-absolute-column  ;
A1-relative-column=
  letter, {letter}  ;  (* see semantic rules at 3.17.2.3.1 *)
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
letter=
  "a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|
  "n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|
  "A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|
  "N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z" ;
A1-absolute-column=
  "$", A1-relative-column  ;
A1-row=
  A1-relative-row  |
  A1-absolute-row  ;
A1-relative-row=
  digit-sequence  ;  (* shall be non-zero; see 3.17.2.3.1 *)
A1-absolute-row=
  “$”, A1-relative-row  ;
```

In this style, each row has a numeric heading numbered sequentially from the top down, starting at 1. Each column has an alphabetic heading named sequentially from left-to-right, A–Z, then AA–AZ, BA–BZ, …, ZA–ZZ, AAA–AAZ, ABA–ABZ, and so on. Column letters are not case-sensitive.

A relative reference to a single cell is written as its column letter immediately followed by its row number. A relative reference to a whole row is written as its row number. A relative reference to a whole column is written as its column letter. A reference to a range of two or more cells is written as two single-cell references separated by the binary range operator (`:`). An absolute A1 reference is made up of a cell's column letter followed by its row number, with each being preceded by a dollar character (`$`). [*Example*: `A2`, `B34`, and `B5:D8` are relative A1 references. `$A$2`, `$B$34`, and `$B$5:$D$8` are absolute A1 references. `$A2`, `B$34`, and `$B5:D$8` are mixed A1 references. *end example*]

[*Example*: `SUM(Sheet2:Sheet13!B5)` adds all the values contained in cell B5 on all the worksheets between and including Sheet2 and Sheet13. *end example*]

For rules on how deal with potential ambiguities between cell references and defined names, see §18.17.5.1.

18.17.2.3.2 R1C1-Style Cell Reference

A cell reference using the R1C1 reference style has the following form:

```
R1C1-reference=
  R1C1-row-only    |
  R1C1-column-only    |
  R1C1-row,  R1C1-column  ;
R1C1-row-only=
  "R",  R1C1-absolute-number,   |
  "R[",  R1C1-relative-number, "]"  ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
R1C1-row=
  R1C1-relative-row  |
  R1C1-absolute-row  ;
R1C1-relative-row=
  "R[",  R1C1-relative-number, "]"  ;
R1C1-absolute-row=
  "R"    |
  "R", R1C1-absolute-number  ;
R1C1-column-only=
  "C",  R1C1-abolute-number  |
  "C[",  R1C1-relative-number,  "]"  ;
R1C1-column=
  R1C1-relative-column  |
  R1C1-absolute-column  ;
R1C1-relative-column=
  "C[",  R1C1-relative-number,  "]"  ;
R1C1-absolute-column=
  "C"    |
  "C", R1C1-absolute-number  ;
R1C1-relative-number=
  ["-"], digit-sequence   ;
R1C1-absolute-number=
  digit-sequence  ;  (* shall be non-zero; see 3.17.2.3.2 *)
```

In this style, each row has a numeric heading numbered sequentially from the top down, starting at 1. Each column has a numeric heading numbered sequentially from left-to-right, starting at 1.

A whole row is referenced by omitting the column, and a whole column is referenced by omitting the row. An absolute row or column reference uses absolute row or column numbers, respectively. A relative row or column reference uses, respectively, row or column offsets from the cell containing the formula, with a negative offset indicating a row to the left or a column above, and a positive offset indicateing a row to the right or a column below. Specifying an offset of zero is equivalent to omitting that offset and its delimiting brackets. [*Example*: `R[-2]C` refers to the cell two rows up and in the same column, `R[2]C[2]` refers to the cell two rows down and two columns to the right, R2C2 refers to the cell in the second row and in the second column, `R[-1]` refers to the entire row above the active cell, and R refers to the current row. *end example*]

The R1C1 alternate reference style can only be used at runtime. See §18.17.6.1 for XML-related details.

18.17.2.4 Functions

A *function* is a named formula that takes zero or more arguments, performs an operation, and, optionally, returns a result. A function call has the following form:

```
function-call=
  function-name,  "(",  [argument-list],  ")"    ;
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
function-name=
  prefixed-function-name       |
  predefined-function-name     |
  user-defined-function-name     ;
predefined-function-name=
  "ABS"  |  "ACOS"  |  "ACOSH"
     |  ( any of the other functions defined in §18.17.7)  ;
prefixed-function-name=
  "ISO.", predefined-function-name |
  "ECMA.", predefined-function-name
user-defined-function-name=
  letter,  [ user-defined-name-characters ]    ;
user-defined-name-characters=
  user-defined-name-character,  {user-defined-name-character}  ;
user-defined-name-character=
  letter  |  decimal-digit  |  full-stop    ;
argument-list=
  argument,  {  comma,  argument  }  ;
comma=
  ","  ;
argument=
  expression  ;
```

*predefined-function-name*s and *user-defined-function-name*s are not case-sensitive.

A *user-defined-function-name* shall not have any of the following forms:

- TRUE or FALSE
- *name*
- *cell-reference*

[*Guidance*: An implementation is encouraged to support *user-defined-function-name*s at least as long as 255 characters. *end guidance*]

The semantics of a call to a function having a *user-defined-function-name* are unspecified.

[*Example*: Here are some function calls: `PI()`, `POWER(A1,B3)`, and `SUM(C6:C10)`. *end example*]

An argument to a function can be a call to a function. That is, function calls can nest. [G*uidance* An implementation is encouraged to support at least 64 levels of nested function calls. *end guidance*]

Some functions take a variable number of arguments. This is indicated in the Syntax sections of §18.17.7 by their having *argument-list* as all, or the trailing part, of their argument list. The total number of arguments that shall be passed to such functions is at least 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Guidance* An implementation is encouraged to support function calls having at least 255 arguments. *end guidance*]

Expressions can have one or more values. Scalar expressions designate a single value, and cell references and array constants can designate multiple values. In the case of a multi-value expression, the way in which this is handled by a function when passed as an argument depends on a number of factors.

Most functions and operators expect either single- or multi-valued arguments and perform all of the array calculations whenever multi-valued arguments are present. [*Example*: `SQRT({1;2;3;4})`; see the examples in §18.17.2. *end example*]

When a function expects a single-valued argument but a multi-valued expression is passed, an attempt can be made to convert that set of values to a single value. For an array value or constant, the value of the expression is the value of the first element within that array value or constant. For a cell range, the first element can be used, or implicit intersection can be performed—the exact behavior is unspecified.

When a function expects a multi-valued argument but a single-valued expression is passed, that single-valued argument is treated as a 1x1 array.

For rules on how deal with potential ambiguities between function names and defined names, see §18.17.5.1.

18.17.2.5 Names

A *name* is an alias for a constant, a cell reference, or a formula. [*Note*: A name in a formula can make it easier to understand the purpose of that formula. For example, the formula `SUM(FirstQuarterSales)` is easier to identify than `SUM(C20:C30)`. *end note*]

Here is the syntax for *name*:

```
name=
  [ workbook-name, "!" ], name-start-character, [ name-characters ]  ;
name-start-character=
  letter  |  underscore  |  backslash  ;
underscore=
  "_"  ;
backslash=
  "\"  ;
name-characters=
  name-character,  {name-character}  ;
name-character=
  letter  |  decimal-digit  |  underscore  |  full-stop  ;
```

*name*s are not case-sensitive.

All *name*s within a workbook shall be unique. If the same *name*s are defined in two workbooks, both *name*s can be used in the same context be prefixing them with their corresponding workbook name and an exclamation

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

mark (`!`). [*Example*: `SUM(Sales.xlsx!ProjectedSales)` refers to the named range `ProjectedSales` in the workbook named `Sales.xlsx`. *end example*]

A *name* shall not have any of the following forms:

- TRUE or FALSE
- *user-defined-function-name*
- *cell-reference*

[*Guidance* An implementation is encouraged to support *name*s at least as long as 255 characters. *end guidance*]

For rules on how deal with potential ambiguities between function names and defined names, or between cell references and defined names, see §18.17.5.1.

18.17.2.6 Types and Values

Each *expression* has a type. SpreadsheetML formulas support the following types: array, error, logical, number, and text.

An array value or constant represents a collection of one or more elements, whose values can have any type (i.e., the elements of an array need not all have the same type).

An error value (§18.17.3) or constant represents an error, and can have any value defined for *error-constant* (§18.17.2.1).

A logical value or constant represents a truth value, and can have any value defined for *logical-constant* (§18.17.2.1).

A numeric value or constant represents a real number, and can have any value defined for *numeric-constant* (§18.17.2.1). The term "number" is used as a generic name for any expression of type *number*.

A text value or constant represents arbitrary text, and can have any value defined for *string-constant* (§18.17.2.1). The term "string" is used as a generic name for any expression of type *text*.

An implementation is permitted to provide an implicit conversion from *string-constant* to number. However, the rules by which such conversions take place are implementation-defined. [*Example*: An implementation might choose to accept "123"+10 by converting the string "123" to the number 123. Such conversions might be locale-specific in that a *string-constant* such as "10,56" might be converted to 10.56 in some locales, but not in others, depending on the radix point character. *end example*]

[*Guidance* An implementation is encouraged to support strings at least as long as 32,767 characters. *end guidance*]

A complex number is represented as a string in one of two equivalent text formats: *x* + *y*`i` or *x* + *y*`j`, where *x* is the real part, and *y* is the imaginary part. [*Example*: `"3+4i"` and `"-2.5-34.6j"` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.2.7 Single- and Array Formulas

A formula can either be a single-cell formula, or an array formula.

A single-cell formula is applied to a single cell while an array formula is applied to a range of cells as a group.

When a single-cell formula results in a single value, the designated cell takes on that value. [*Example*: When cell `A10` contains `SIN(0.3)`, the result stored in that cell is `0.295520207`. *end example*]

When a single-cell formula results in multiple values, the designated cell takes on the first of those values. [*Example*: When cell A10 contains `SIN({0.3,0.4,0.5})`, the result stored in that cell is 0.295520207 (SIN(0.3)). *end example*]

Array formulas are an extension of the formula paradigm. They allow for combining of several formula operations into one, and the returning of multiple results. Array formulas span one or more cells. There are three primary functions that array formulas perform:

5. Returning multiple results from one single function call. [*Example*: the function `LINEST` returns an array of several results, and, therefore, ought to be spanned across several cells as an array formula in order to display those results. *end example*]
6. Executing one formula several times to generate an array of results. [*Example*: The array formula `SIN(A1:A3)` returns an array with three elements, those being the sines of A1, A2 and A3, respectively. *end example*]
7. Returning one single result from an operation that incorporated arrays as an intermediate step. [*Example*: The array formula `SUM(SIN(A1:A3))` returns a single result, that being the SUM of the sines of A1, A2 and A3, respectively. *end example*]

When an array formula results in a single value, all of the cells spanned by the array formula take on that value. [*Example*: When the group of cells `A10:A12` contains `SIN(0.3)`, the result stored in each of those cells is `0.295520207`. *end example*]

When an array formula results in multiple values, the designated cells take on corresponding values, according to the shape of the cell group and the values. Specifically,

- If the cell group and values have the same shape (i.e., the same number of rows and columns), each cell takes on the value corresponding to its relative position.
- If the cell group has fewer columns than the values, the left-most columns of the values are stored in the cells.
- If the cell group has fewer rows than the values, the top-most rows of the values are stored in the cells.
- If the cell group has more columns than the values, each cell takes on the value corresponding to its relative position, except that
    - For a cell group 1x*N* array or a two-dimensional array, the excess right-most cells take on an unspecified value.
    - For a cell group *N*x1 array, the excess columns are clones of the first column.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- If the cell group has more rows than the values, each cell takes on the value corresponding to its relative position, except that:
  - For a cell group *N*x1 array or a two-dimensional array, the excess bottom-most cells take on an unspecified value.
  - For a cell group 1x*N* array, the excess rows are clones of the first row.

[*Example*: Case 1: The 1x3 group of cells `A20:C20` has applied to it the array formula `SIN({0.3,0.4,0.5})`. The number of rows and columns in the group exactly matches the number of rows and columns in the result. Those cells then contain 0.295520207, 0.389418342, and 0.479425539, which correspond to `SIN(0.3)`, `SIN(0.4)`, and `SIN(0.5)`, respectively.

Case 2: The 1x2 group of cells `A20:B20` has applied to it the array formula `SIN({0.3,0.4,0.5})`. The number of columns in the group is less than the number of columns in the result. (The number of rows is the same in each.) Those cells then contain 0.295520207 and 0.389418342, which correspond to `SIN(0.3)` and `SIN(0.4)`, respectively, the left-most part of the set of values.

Case 3: The 1x4 group of cells `A20:D20` has applied to it the array formula `SIN({0.3,0.4,0.5})`. The number of columns in the group is greater than the number of columns in the result. (The number of rows is the same in each.) Those cells then contain 0.295520207, 0.389418342, 0.479425539, and an unspecified value, which correspond to `SIN(0.3)`, `SIN(0.4)`, and `SIN(0.5)`, respectively, with the fourth value being unspecified.

Case 4: The 2x2 group of cells `A30:B31` has applied to it the array formula `SIN({0.1,0.2,0.3})`. The number of columns in the group is less than the number of columns in the result. As a result, the cells in row 30 contain 0.295520207 and 0.389418342, which correspond to `SIN(0.3)` and `SIN(0.4)`, respectively. The number of rows in the group is greater than the number of rows in the result, so the cells in 31 are a copy of the cells in row 30. The left-most part of the set of values is propagated into the cells.

Case 5: The 2x2 group of cells `A40:B41` has applied to it the array formula `SIN({0.1,0.2,0.3; 0.4,0.5,0.6; 0.7,0.8,0.9})`. The number of columns in the group is less than the number of columns in the result. As a result, the left-most column values are stored. The number of rows in the group is less than the number of rows in the result. As a result, the top-most column values are stored. *end example*]

18.17.3 Error values

The evaluation of an expression can result in an error having one of a number of *error value*s. These error values are:

Error Value	Reason for Occurrence
#DIV/0!	Intended to indicate when any number (including zero) or any error code is divided by zero.
#GETTING_DATA	Intended to indicate when a cell reference cannot be evaluated because the value for the cell has not been retrieved or calculated. [*Note*: This can happen when connected to an OLAP cube. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Error Value	Reason for Occurrence
	This error constant differs from #N/A in that #GETTING_DATA is used when there is an expectation that the value for the cell will eventually be available, whereas #N/A is used when there is no such expectation.
`#N/A`	Intended to indicate when a designated value is not available. [*Example*: Some functions, such as `SUMX2MY2`, perform a series of operations on corresponding elements in two arrays. If those arrays do not have the same number of elements, then for some elements in the longer array, there are no corresponding elements in the shorter one; that is, one or more values in the shorter array are not available. *end example*] This error value can be produced by calling the function `NA` (§18.17.7.223).
`#NAME?`	Intended to indicate when what looks like a name is used, but no such name has been defined. [*Example*: `XYZ/3`, where XYZ is not a defined name. `Total is & A10`, where neither `Total` nor `is` is a defined name. Presumably, `"Total is " & A10` was intended. `SUM(A1C10)`, where the range `A1:C10` was intended. *end example*]
`#NULL!`	Intended to indicate when two areas are required to intersect, but do not. [*Example*: In the case of `SUM(B1 C1)`, the space between B1 and C1 is treated as the binary intersection operator, when a comma was intended. *end example*]
`#NUM!`	Intended to indicate when an argument to a function has a compatible type, but has a value that is outside the domain over which that function is defined. (This is known as a *domain error*.) [*Example*: Certain calls to `ASIN`, `ATANH`, `FACT`, and `SQRT` might result in domain errors. *end example*] Intended to indicate that the result of a function cannot be represented in a value of the specified type, typically due to extreme magnitude. (This is known as a *range error*.) [*Example*: `FACT(1000)` might result in a range error. *end example*]
`#REF!`	Intended to indicate when a cell reference cannot be evaluated. [*Example*: If a formula contains a reference to a cell, and then the row or column containing that cell is deleted, a `#REF!` error results. If a worksheet does not support 20,001 columns, `OFFSET(A1,0,20000)` will result in a `#REF!` error. *end example*]
`#VALUE!`	Intended to indicate when an incompatible type argument is passed to a function, or an incompatible type operand is used with an operator. [*Example*: In the case of a function argument, text was expected, but a number was provided *end example*]

Each error value has a corresponding *error-constant* (§18.17.2.1).

[*Note*: A number of functions operate on error values: They include `ERROR.TYPE` (§18.17.7.110), `ISERR` (§18.17.7.175), `ISERROR` (§18.17.7.176), and `ISNA` (§18.17.7.179). *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.4 Dates and Times

Dates and times in cells in SpreadsheetML are stored as strings, using the ISO 8601 lexical formats defined below.

The earliest date permitted is 0001-01-01, 00:00. The latest date permitted is 9999-12-31, 23:59:59.999. The time midnight shall be expressed always with hour component 0 and not with hour component 24. Leap seconds are not permitted – the maximum number of seconds expressed in a minute shall be 60.

Values with only a date component shall be expressed using the Complete, Extended Format Calendar Date representation, as defined in ISO 8601, §B.1.1 and §B.2.1.

[*Example*: The date 5 October 1975 is expressed in SpreadsheetML as

1975-10-05

end example]

Values with only a time-of-day component shall be expressed using the Complete, Extended Format Time Of Day representation, as defined in ISO 8601, §B.1.2 and §B.2.2. The decimal separator shall be a full stop (period), and fractional seconds should be expressed with no more than three decimal places.

[*Rationale*: There are significant differences among standards' and systems' support for fractional seconds in time values. Allowing implementations to choose a level of precision support that is most appropriate to each implementation provides the most flexibility for different usage scenarios. A recommended baseline precision sets a goal for support with the intention of improving interoperability. *end rationale*]

[*Guidance*: Implementations are encouraged to document their time precision support to enhance interoperability. *end guidance*]

[*Example*: The time-of-day 08:30 can be expressed in the following ways within SpreadsheetML:

08:30
08:30:00
08:30:00.000
end example]

Values with both date and time-of-day components shall be expressed using the Complete, Extended Format Calendar Date and Time Of Day representation, as defined in ISO 8601, §B.1.3 and §B.2.3. For the time component, only seconds may use a decimal separator, the decimal separator shall be a full stop (period) and fractional seconds should be expressed with no more than three decimal places.

[*Example*: The date 22 November 1976 at local time 08:30 can be expressed in the following ways within SpreadsheetML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

1976-11-22T08:30
1976-11-22T08:30:00
1976-11-22T08:30:00.000

The date 15 October 1582—the day the Gregorian calendar went into effect for some countries—can be expressed in the following ways:

1582-10-15
1582-10-15T00:00
1582-10-15T00:00:00
1582-10-15T00:00:00.000

end example]

[*Note:* SpreadsheetML relates all dates to the proleptic Gregorian calendar of ISO 8601, treating time periods extending into the past and into the distant future as if the Gregorian calendar is in effect for all of those days. January 1 is always the first day of each year, ignoring historical changes to the period of the calendar year. The gaps and shifts introduced as part of calendar reforms and for introduction of leap seconds are ignored under the proleptic Gregorian calendar system. *end note*]

Wherever a calculation in a formula is specified to apply to number values and a date or time is provided, the effect shall be the same as if the date and/or time value is converted to the corresponding serial date-time. Wherever a calculation in a formula is specified to apply to or to deliver a date and/or time value, and a number value is supplied, the number value is interpreted as a serial date-time for the date and/or time. The relationships between serial date-times and dates and times are specified in §18.17.4.1, §18.17.4.2, and §18.17.4.3.

18.17.4.1 Date Conversion for Serial Date-Times

A serial date-time is a number that represents a date and time. This signed value is in units of days relative to the base date for the selected date system. Serial date-times increase by 1 into each successive day and decrease by 1 into each preceding day. Fractional portions of serial date-times represent fractions of a single day. [*Example*: When using the 1900 date system, which has a base date of 30th December 1899, a serial date-time of 1.5 represents midday on the 31st December 1899 (serial date-time day 1), or 1899-12-31T12:00. A serial date-time of -4.25 represents 6 pm on the 25th December 1899, or 1899-12-25T18:00. *end example*] The base dates and the related serial date-times represent local date and time.

Two different bases are used for converting dates to and from serial date-times:

- In the *1900 date system*, the lower limit is January 1st, 0001 00:00:00, which has a serial date-time of -693593. The upper-limit is December 31st, 9999, 23:59:59.999, which has a serial date-time of 2,958,465.9999884. The base date for this system is 00:00:00 on December 30th, 1899, which has a serial date-time of 0.
- In the *1904 date system*, the lower limit is January 1st, 0001, 00:00:00, which has a serial date-time of -695055. The upper limit is December 31st, 9999, 23:59:59.999, which has a serial date-time of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

2,957,003.9999884. The base date for this system is 00:00:00 on January 1st, 1904, which has a serial date-time of 0.

A serial date-time outside the temporal range for the selected date system is invalid.

The date system is specified by the value of the date1904 attribute of the workbookPr element. [*Example*:

1900 date system: `<workbookPr showObjects="all"/>`
1904 date system: `<workbookPr date1904="1" showObjects="all"/>`

end example]

18.17.4.2 Time Conversion for Serial Date-Times

Time of day is represented by a serial date-time less than 1, but not less than 0. Values from 0–0.99999999 represent times from *the* starting instant 0:00:00 (12:00:00 AM) to the last instant 23:59:59 (11:59:59 P.M.).

For any serial date-time, the serial time-of-day is the serial date-time minus the serial date-time of the day in which the time-of-day occurs. The serial date-time of the day in which a serial date-time occurs is the greatest integer that does not exceed the serial date-time.

[*Example:*

The serial date-time 4.66666667 is in serial day 4, and the time-of-day serial date-time is 4.66666667 – 4, which is 0.66666667.

The serial date-time -2.00000001 is in serial day -3, and the time-of-day serial date-time is -2.00000001 – (-3), which is 0.99999999.

end example]

Going forward in time, the time component of a serial date-time increases by 1/86,400 each second. [*Note*: As such, the time 12:00 has a serial date-time time component of 0.5. *end note*]

[*Example*:

The serial date-time `0.0000000`… represents `00:00:00`
The serial date-time `0.0000115`… represents `00:00:01`
The serial date-time `0.4207639`… represents `10:05:54`
The serial date-time `0.5000000`… represents `12:00:00`
The serial date-time `0.9999884`… represents `23:59:59`

end example]

18.17.4.3 Combined Date and Time Conversion for Serial Date-Times

The serial date-time corresponding to a date component can be added to any serial date-time for a time-of-day component to determine the serial date-time for the combined date-time.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: In the 1900 date system, the serial date-time -1.25 represents December 28, 1899, 18:00. *end note*]

[*Example*: For the 1900 date system:

The serial date-time -2337.999989… represents 1893-08-05T00:00:01
The serial date-time `3687.4207639`… represents 1910-02-03T10:05:54
The serial date-time `2.5000000`… represents 1900-01-01T12:00:00
The serial date-time `2958465.9999884`… represents 9999-12-31T23:59:59

For the 1904 date system:

The serial date-time -3799.999989… represents 1893-08-05T00:00:01
The serial date-time `2225.4207639`… represents 1910-02-03T10:05:54
The serial date-time `0.5000000`… represents 1904-01-01T12:00:00
The serial date-time `2957003.9999884`… represents 9999-12-31T23:59:59

end example]

18.17.5 Limits and Precision

18.17.5.1 Limits

In SpreadsheetML, cell references range from column A1–A1048576 (column A:A) to column XFD1–XFD1048576 (column XFD:XFD).

An implementation can extend this range. However, to avoid ambiguities, it is necessary to ensure that defined names are distinct from cell references, or that one takes precedence over the other. With this in mind, the following rules apply:

- A producer or consumer shall consider a defined name of the form used by cells in the range A1–XFD1048576 to be an error.
- All other names outside this range can be defined names and shall override a cell reference if an ambiguity exists.

[*Example*: `LOG10` is always a cell reference, `LOG10(…)` is always a formula, and `LOGO1000` can be a defined name that overrides a cell reference. *end example*]

18.17.5.2 Precision

In order to clarify the semantics of cell formulas and values in SpreadsheetML, it is necessary to specify the precision of the numbers being represented in the file format. These numbers are therefore regarded as ranging over a specific value space, which defaults to the following:

The *value space* consists of the values (-1)^s × m × 2^n, where s is 0 or 1, m is an integer greater than or equal to 0 and less than 2^53, and n is an integer between -1074 and 971, inclusive. m is herein referred to as the *binary*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

mantissa, and n is herein referred to as the *binary exponent*. [*Note*: The default precision is patterned after the IEC double-precision 64-bit floating-point type [IEC 60559]. *end note*]

Implementing applications can use the characteristics markup (§22.7) to specify other value spaces to replace the default in a given workbook. When present in the workbook, the value space defined using the characteristics markup overrides the default value space.

Regardless of the specific value space in use, values shall have a lexical representation as described in §18.17.5.3. Any numerical expression conforming to this lexical description is permitted. However, numbers of higher precision than available in the value space, and numbers that lie outside the range representable in the value space shall be handled as prescribed in §18.17.5.4.

18.17.5.3 Lexical Representation

The value space shall have a lexical representation consisting of a base 10 mantissa followed, optionally, by the character "E" or "e", followed by a base 10 exponent. The exponent shall be an integer. The mantissa shall be a decimal number. The representations for exponent and mantissa shall follow the lexical rules for integer and decimal below. If the "E" or "e" and the following exponent are omitted, an exponent value of 0 is assumed.

Lexical representations for zero can take a positive or negative sign.

[*Example*: -1E4, 1267.43233E12, 12.78e-2, 12 , -0, and 0 are all literals for numbers in the default value space. 4503599627370497.5 is also a literal, although it represents the same value as 4503599627370497 (2^52 + 1) in the default value space (as explained in §18.17.5.4). *end example*]

An *Integer* has a lexical representation consisting of a finite-length sequence of decimal digits (#x30–#x39) with an optional leading sign. If the sign is omitted, "+" is assumed. [*Example*: -1, 0, 12678967543233, +100000. *end example*]

A *Decimal Number* has a lexical representation consisting of a finite-length sequence of decimal digits (#x30–#x39) separated by a period as a decimal indicator. An optional leading sign is allowed. If the sign is omitted, "+" is assumed. Leading and trailing zeroes are optional. If the fractional part is zero, the period and following zero(s) can be omitted. [*Example*: -1.23, 12678967.543233, +100000.00, 210. *end example*]

18.17.5.4 Interpretation

Strings that are permitted according to the lexical definition in §18.17.5.3 shall be interpreted as values in the value space as follows:

1. The mantissa shall be interpreted as a real number expressed in base 10
2. The exponent shall be interpreted as an integer expressed in base 10
3. The raw value for a numerical expression shall be interpreted as mantissa x $10^{exponent}$
4. If the absolute value is larger than the largest value in the value space (2^1024 minus 2^971) then a consuming application shall treat this as equivalent to the error value `#NUM!` (§18.17.3). Otherwise the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

value in the value space that is closest to the raw value is chosen as the interpretation. In the case that two values are equally close, the one with the smaller absolute value is chosen.

18.17.6 XML Representation

18.17.6.1 Cell Reference Style

A workbook saved with reference style A1 (§18.17.2.3.1), shall have the refMode attribute of the calcPr element (§18.2.2) in the Workbook part's XML omitted or set to A1. A workbook saved with reference style R1C1 (§18.17.2.3.2) shall have that refMode attribute set to R1C1. [*Example*: With R1C1 mode set, here is how the XML might look:

```
<workbook …>
  …
  <calcPr calcId="122211" fullCalcOnLoad="1" refMode="R1C1"/>
  …
</workbook>
```

end example]

Regardless of the value of the refMode attribute, cell references shall be stored in XML in the A1 form. This attribute's value tells an implementation which reference style to use at runtime.

18.17.6.2 Scalar Formulas

A scalar formula shall be represented in a worksheet's XML by an f element that contains the text of the formula, and a v element that contains the text version of the last computed value for that formula. This pair of elements shall be inside a c element, which is, in turn, shall be inside a row element. [*Example*: Consider the scalar formula SQRT(C2^2+D2^2), where C2 refers to a cell containing the number 12.5, and D2 refers to a cell containing the number 9.6. The corresponding XML might be as follows:

```
<row r="2" spans="2:4">
  <c r="B2" s="40">
    <f>SQRT(C2^2+D2^2)</f>
    <v>15.761027885261798</v>
  </c>
  <c r="C2" s="0">
    <v>12.5</v>
  </c>
  <c r="D2" s="0">
    <v>9.6</v>
  </c>
</row>
```

In the scalar formula CONCATENATE("The total is ",C7," units"), C7 refers to a cell containing the number 23. The corresponding XML might be as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<row r="7" spans="2:4" ht="285">
  <c r="B7" s="4" t="str">
    <f>CONCATENATE("The total is ",C7," units")</f>
    <v>The total is 23 units</v>
  </c>
  <c r="C7" s="0">
    <v>23</v>
  </c>
</row>
```

As the function CONCATENATE returns a string, the value for the cell's t attribute is `str`.

end example]

18.17.6.3 Array Formulas

An array formula shall be represented in XML just like other formulas, except that the array formula's f element shall contain an attribute t, whose value shall be `array`.

For a single-cell formula, the r attribute shall designate that cell. [*Example*: Consider the array formula `SUM(C11:C12*D11:D12)`. The corresponding XML might be as follows:

```
<row r="11" spans="2:4" ht="300">
  <c r="B11" s="16">
    <f t="array" r="B11">SUM(C11:C12*D11:D12)</f>
    <v>110</v>
  </c>
  <c r="C11" s="4">
    <v>10</v>
  </c>
  <c r="D11" s="0">
    <v>3</v>
  </c>
</row>
<row r="12" spans="2:4" ht="285">
  <c r="C12" s="4">
    <v>20</v>
  </c>
  <c r="D12" s="0">
    <v>4</v>
  </c>
</row>
```

As this formula is a single-cell formula, the r attribute contains the name of that cell, B11. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

For an array formula spanning multiple cells, the r attribute of the top-left cell of the range of cells to which that formula applies shall designate the range of cells to which that formula applies. The c elements for all cells except the top-left cell in that range shall not have an f element; however, they shall each have a v element. [*Example*: Consider the array formula A1:A3*B1:B3, which is applied to the cell range C1:C3. The corresponding XML might be as follows:

```
<row r="1" spans="1:3">
  <c r="A1" s="0">
    <v>112</v>
  </c>
  <c r="B1" s="0">
    <v>2.34</v>
  </c>
  <c r="C1" s="0">
    <f t="array" r="C1:C3">A1:A3*B1:B3</f>
    <v>262.08</v>
  </c>
</row>
<row r="2" spans="1:3">
  <c r="A2" s="0">
    <v>209</v>
  </c>
  <c r="B2" s="0">
    <v>1.28</v>
  </c>
  <c r="C2" s="0">
    <v>267.52</v>
  </c>
</row>
<row r="3" spans="1:3">
  <c r="A3" s="0">
    <v>128</v>
  </c>
  <c r="B3" s="0">
    <v>3.12</v>
  </c>
  <c r="C3" s="0">
    <v>399.36</v>
  </c>
</row>
```

As this formula is an array formula spanning multiple cells, the r attribute of cell C1 contains the name of that cell range, C1:C3, and cells C2 and C3 do not have an f element. *end example*]

18.17.6.4 Formula Evaluation Order

The order in which formulas are evaluated is determined by the order of their corresponding c elements in the calcChain element of the Calculation Chain part (§18.6.2).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.17.6.5 Name Representation

A formula can contain one or more names. These names shall be defined in the Workbook part's XML with each being the subject of a definedName element, inside a definedNames element. [*Example*: Consider the scalar formula `SUM(value1,value2)`. The corresponding XML might be as follows:

```
<definedNames>
  <definedName name="value1" localSheetId="0">Sheet2!$B$2</definedName>
  <definedName name="value2" localSheetId="0">Sheet2!$B$3</definedName>
</definedNames>
…
<c r="E5" s="0">
  <f ce="1">SUM(value1,value2)</f>
  <v>8</v>
</c>
```

end example]

Each name shall be the subject of an lpstr element in the Application-Defined File Properties part.

```
<TitlesOfParts>
  <vt:vector … baseType="lpstr">
    <vt:lpstr>Sheet1</vt:lpstr>
    <vt:lpstr>Sheet2</vt:lpstr>
    <vt:lpstr>Sheet3</vt:lpstr>
    <vt:lpstr>value1</vt:lpstr>
    <vt:lpstr>value2</vt:lpstr>
  </vt:vector>
</TitlesOfParts>
```

18.17.6.6 Value Representation

The most recent value of a formula shall be stored in the corresponding v element, as follows:

Result Type	Representation
array	The text form of the array's value.
error	The text form of the error value.
logical	The text `0` for FALSE and `1` for TRUE.
number	The unformatted text form of the number, as accurately as possible.
text	All of the characters in the text.

18.17.6.7 Dates and Times

When a SpreadsheetML cell contains a date-time, the value of the cell is expressed as a string conforming to one of the ISO 8601 lexical formats specified in §18.17.4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

```
<c t="d">
  <f>DATE(1582,10,15)+0.5</f>
    <v>1582-10-15T12:00</v>
  </c>
```

end example]

18.17.7 Predefined Function Definitions

Each of the subclauses below this subclause describes a separate function, and each description contains a section marked **Syntax**. That section contains pieces of the function grammar as they pertain to that specific function. These pieces are presented in a slightly simpler form to aid in the understanding of the description. In those sections, the left-square-bracket and right-square-bracket designate the start and end of an option, as used in ISO/IEC 14977. However, the function-name, the open-parenthesis and the close-parenthesis designate actual literal text, as does each comma. [*Note*: Therefore, in a strict presentation according to ISO/IEC 14977, each would appear with double-quotes surrounding each instance. *end note*]

The **Syntax** section for each function defined in this subclause corresponds to a call to that function. Except for *argument-list*, the names in any **Syntax** section typeset as in *number* and *string-1*, are parameter names for that function, and are local to that function definition's description. *argument-list* is the name of a production in the grammar, and, as defined in §18.17.2, permits a comma-separated list of *argument*s.

When the type of an argument passed to a function is incompatible with the type expected the error value `#VALUE!` is returned by that function.

The set of predefined functions is divided into the following functional categories [*Note*: The predefined functions defined here reflect current spreadsheet semantics and might not match common practice in other contexts. New functions might be added in future versions of the specification. *end note*]:

All predefined functions can be used with their simple name, with the prefix `ECMA.` or with the prefix `ISO.`, with the following exception: the predefined function named `CEILING` in ISO/IEC 29500 can only be used with the prefix `ECMA.`. The predefined function named `ISO.CEILING` is specified in ISO/IEC 29500.

Category	Formulas
Cube	CUBEKPIMEMBER (§18.17.7.65), CUBEMEMBER (§18.17.7.66), CUBEMEMBERPROPERTY (§18.17.7.67), CUBERANKEDMEMBER (§18.17.7.68), CUBESET (§18.17.7.69), CUBESETCOUNT (§18.17.7.70), CUBEVALUE (§18.17.7.71)
Database	DAVERAGE (§18.17.7.77), DCOUNT (§18.17.7.81), DCOUNTA (§18.17.7.82), DGET (§18.17.7.90), DMAX (§18.17.7.92), DMIN (§18.17.7.93), DPRODUCT (§18.17.7.97), DSTDEV (§18.17.7.98), DSTDEVP (§18.17.7.99), DSUM (§18.17.7.100), DVAR (§18.17.7.102), and DVARP (§18.17.7.103).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Category	Formulas
Date and Time	DATE (§18.17.7.74), DATEDIF (§18.17.7.75), DATEVALUE (§18.17.7.76), DAY (§18.17.7.78), DAYS360 (§18.17.7.79), EDATE (§18.17.7.105), EOMONTH (§18.17.7.107), HOUR (§18.17.7.144), MINUTE (§18.17.7.214), MONTH (§18.17.7.220), NETWORKDAYS (§18.17.7.226), NETWORKDAYS.INTL (§18.17.7.227), NOW (§18.17.7.234), SECOND (§18.17.7.287), TIME (§18.17.7.323), TIMEVALUE (§18.17.7.324), TODAY (§18.17.7.326), WEEKDAY (§18.17.7.344), WEEKNUM (§18.17.7.345), WORKDAY (§18.17.7.347) , WORKDAY.INTL (§18.17.7.348), YEAR (§18.17.7.351), and YEARFRAC (§18.17.7.352)
Engineering	BESSELI (§18.17.7.23), BESSELJ (§18.17.7.24), BESSELK (§18.17.7.25), BESSELY (§18.17.7.26), BIN2DEC (§18.17.7.29), BIN2HEX (§18.17.7.30), BIN2OCT (§18.17.7.31), COMPLEX (§18.17.7.45), CONVERT (§18.17.7.48), DEC2BIN (§18.17.7.84), DEC2HEX (§18.17.7.85), DEC2OCT (§18.17.7.86), DELTA (§18.17.7.88), ERF (§18.17.7.108), ERFC (§18.17.7.109), GESTEP (§18.17.7.136), HEX2BIN (§18.17.7.140), HEX2DEC (§18.17.7.141), HEX2OCT (§18.17.7.142), IMABS (§18.17.7.149), IMAGINARY (§18.17.7.150), IMARGUMENT (§18.17.7.151), IMCONJUGATE (§18.17.7.152), IMCOS (§18.17.7.153), IMDIV (§18.17.7.154), IMEXP (§18.17.7.155), IMLN (§18.17.7.156), IMLOG10 (§18.17.7.157), IMLOG2 (§18.17.7.158), IMPOWER (§18.17.7.159), IMPRODUCT (§18.17.7.160), IMREAL (§18.17.7.161), IMSIN (§18.17.7.162), IMSQRT (§18.17.7.163), IMSUB (§18.17.7.164), IMSUM (§18.17.7.165), OCT2BIN (§18.17.7.237), OCT2DEC (§18.17.7.238), and OCT2HEX (§18.17.7.239).
Financial	ACCRINT (§18.17.7.2), ACCRINTM (§18.17.7.3), AMORDEGRC (§18.17.7.7), AMORLINC (§18.17.7.8), COUPDAYBS (§18.17.7.57), COUPDAYS (§18.17.7.58), COUPDAYSNC (§18.17.7.59), COUPNCD (§18.17.7.60), COUPNUM (§18.17.7.61), COUPPCD (§18.17.7.62), CUMIPMT (§18.17.7.72), CUMPRINC (§18.17.7.73), DB (§18.17.7.80), DDB (§18.17.7.83), DISC (§18.17.7.90), DOLLARDE (§18.17.7.95), DOLLARFR (§18.17.7.96), DURATION (§18.17.7.101), EFFECT (§18.17.7.106), FV (§18.17.7.129), FVSCHEDULE (§18.17.7.130), INTRATE (§18.17.7.171), IPMT (§18.17.7.172), IRR (§18.17.7.173), ISPMT (§18.17.7.184), MDURATION (§18.17.7.208), MIRR (§18.17.7.216), NOMINAL (§18.17.7.228), NPER (§18.17.7.235), NPV (§18.17.7.236), ODDFPRICE (§18.17.7.241), ODDFYIELD (§18.17.7.242), ODDLPRICE (§18.17.7.243), ODDLYIELD (§18.17.7.244), PMT (§18.17.7.253), PPMT (§18.17.7.256), PRICE (§18.17.7.257), PRICEDISC (§18.17.7.258), PRICEMAT (§18.17.7.259), PV (§18.17.7.263), RATE (§18.17.7.270), RECEIVED (§18.17.7.271), SLN (§18.17.7.293), SYD (§18.17.7.314), TBILLEQ (§18.17.7.318), TBILLPRICE (§18.17.7.319), TBILLYIELD (§18.17.7.320), VDB (§18.17.7.342), XIRR (§18.17.7.349), XNPV (§18.17.7.350), YIELD (§18.17.7.353), YIELDDISC (§18.17.7.354), and YIELDMAT (§18.17.7.355).
Information	CELL (§18.17.7.34), ERROR.TYPE (§18.17.7.110), INFO (§18.17.7.168), ISBLANK (§18.17.7.174), ISERR (§18.17.7.175), ISERROR (§18.17.7.176), ISEVEN (§18.17.7.177), ISLOGICAL (§18.17.7.178), ISNA (§18.17.7.179), ISNONTEXT (§18.17.7.180), ISNUMBER (§18.17.7.181), ISODD (§18.17.7.183), ISREF

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Category	Formulas
	(§18.17.7.185), ISTEXT (§18.17.7.186), N (§18.17.7.223), NA (§18.17.7.224), and TYPE (§18.17.7.334).
Logical	AND (§18.17.7.9), FALSE (§18.17.7.117), IF (§18.17.7.147), IFERROR (§18.17.7.148), NOT (§18.17.7.233), OR (§18.17.7.246), and TRUE (§18.17.7.330).
Lookup and Reference	ADDRESS (§18.17.7.6), AREAS (§18.17.7.10), CHOOSE (§18.17.7.39), COLUMN (§18.17.7.42), COLUMNS (§18.17.7.43, GETPIVOTDATA (§18.17.7.137), HLOOKUP (§18.17.7.143), HYPERLINK (§18.17.7.145), INDEX (§18.17.7.166), INDIRECT (§18.17.7.167), LOOKUP (§18.17.7.202), MATCH (§18.17.7.204), OFFSET (§18.17.7.245), ROW (§18.17.7.281), ROWS (§18.17.7.282), RTD (§18.17.7.284), TRANSPOSE (§18.17.7.327), and VLOOKUP (§18.17.7.343).
Math and Trig	ABS (§18.17.7.1), ACOS (§18.17.7.4), ACOSH (§18.17.7.5), ASIN (§18.17.7.12), ASINH (§18.17.7.13), ATAN (§18.17.7.14), ATAN2 (§18.17.7.15), ATANH (§18.17.7.16), CEILING (§18.17.7.33), COMBIN (§18.17.7.44), COS (§18.17.7.50), COSH (§18.17.7.51), DEGREES (§18.17.7.87), ECMA.CEILING (§18.17.7.104), EVEN (§18.17.7.111), EXP (§18.17.7.113), FACT (§18.17.7.115), FACTDOUBLE (§18.17.7.116), FLOOR (§18.17.7.125), GCD (§18.17.7.134), INT (§18.17.7.169), ISO.CEILING (§18.17.7.182), LCM (§18.17.7.190), LN (§18.17.7.196), LOG (§18.17.7.197), LOG10 (§18.17.7.198), MDETERM (§18.17.7.207), MINVERSE (§18.17.7.215), MMULT (§18.17.7.217), MOD (§18.17.7.218), MROUND (§18.17.7.221), MULTINOMIAL (§18.17.7.222), ODD (§18.17.7.240), PI (§18.17.7.252), POWER (§18.17.7.255), PRODUCT (§18.17.7.261), QUOTIENT (§18.17.7.265), RADIANS (§18.17.7.266), RAND (§18.17.7.267), RANDBETWEEN (§18.17.7.268), ROMAN (§18.17.7.277), ROUND (§18.17.7.278), ROUNDDOWN (§18.17.7.279), ROUNDUP (§18.17.7.280), SERIESSUM (§18.17.7.288), SIGN (§18.17.7.289), SIN (§18.17.7.290), SINH (§18.17.7.291), SQRT (§18.17.7.296), SQRTPI (§18.17.7.297), SUBTOTAL (§18.17.7.305), SUM (§18.17.7.306), SUMIF (§18.17.7.307), SUMIFS (§18.17.7.308), SUMPRODUCT (§18.17.7.309), SUMSQ (§18.17.7.310), SUMX2MY2 (§18.17.7.311), SUMX2PY2 (§18.17.7.312), SUMXMY2 (§18.17.7.313), TAN (§18.17.7.316), TANH (§18.17.7.317), and TRUNC (§18.17.7.332).
Statistical	AVEDEV (§18.17.7.17), AVERAGE (§18.17.7.18), AVERAGEA (§18.17.7.19),AVERAGEIF (§18.17.7.20), AVERAGEIFS (§18.17.7.21), BETADIST (§18.17.7.27), BETAINV (§18.17.7.28), BINOMDIST (§18.17.7.32), CHIDIST (§18.17.7.36), CHIINV (§18.17.7.37), CHITEST (§18.17.7.38), CONFIDENCE (§18.17.7.47), CORREL (§18.17.7.49), COUNT (§18.17.7.52), COUNTA (§18.17.7.53), COUNTBLANK (§18.17.7.54), COUNTIF (§18.17.7.55),COUNTIFS (§18.17.7.56), COVAR (§18.17.7.63), CRITBINOM (§18.17.7.64), DEVSQ (§18.17.7.89), EXPONDIST (§18.17.7.114), FDIST (§18.17.7.118), FINV (§18.17.7.121), FISHER (§18.17.7.122), FISHERINV (§18.17.7.123), FORECAST (§18.17.7.126), FREQUENCY (§18.17.7.127), FTEST (§18.17.7.128), GAMMADIST (§18.17.7.131), GAMMAINV (§18.17.7.132), GAMMALN (§18.17.7.133), GEOMEAN (§18.17.7.135), GROWTH (§18.17.7.138), HARMEAN (§18.17.7.139), HYPGEOMDIST (§18.17.7.146), INTERCEPT

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Category	Formulas
	(§18.17.7.170), KURT (§18.17.7.188), LARGE (§18.17.7.189), LINEST (§18.17.7.195), LOGEST (§18.17.7.199), LOGINV (§18.17.7.200), LOGNORMDIST (§18.17.7.201), MAX (§18.17.7.205), MAXA (§18.17.7.206), MEDIAN (§18.17.7.209), MIN (§18.17.7.212), MINA (§18.17.7.213), MODE (§18.17.7.219), NEGBINOMDIST (§18.17.7.225), NORMDIST (§18.17.7.229), NORMINV (§18.17.7.230), NORMSDIST (§18.17.7.231), NORMSINV (§18.17.7.232), PEARSON (§18.17.7.247), PERCENTILE (§18.17.7.248), PERCENTRANK (§18.17.7.249), PERMUT (§18.17.7.250), POISSON (§18.17.7.254), PROB (§18.17.7.260), QUARTILE (§18.17.7.264), RANK (§18.17.7.269), RSQ (§18.17.7.283), SKEW (§18.17.7.292), SLOPE (§18.17.7.294), SMALL (§18.17.7.295), STANDARDIZE (§18.17.7.298), STDEV (§18.17.7.299), STDEVA (§18.17.7.300), STDEVP (§18.17.7.301), STDEVPA (§18.17.7.302), STEYX (§18.17.7.303), TDIST (§18.17.7.321), TINV (§18.17.7.325), TREND (§18.17.7.328), TRIMMEAN (§18.17.7.330), TTEST (§18.17.7.333), VAR (§18.17.7.338), VARA (§18.17.7.339), VARP (§18.17.7.340), VARPA (§18.17.7.341), WEIBULL (§18.17.7.346), and ZTEST (§18.17.7.356).
Text and Data	ASC (§18.17.7.11), BAHTTEXT (§18.17.7.22), CHAR (§18.17.7.35), CLEAN (§18.17.7.40), CODE (§18.17.7.41), CONCATENATE (§18.17.7.46), DOLLAR (§18.17.7.94), EXACT (§18.17.7.112), FIND (§18.17.7.119), FINDB (§18.17.7.120), FIXED (§18.17.7.124), JIS (§18.17.7.187), LEFT (§18.17.7.191), LEFTB (§18.17.7.192), LEN (§18.17.7.193), LENB (§18.17.7.194), LOWER (§18.17.7.203), MID (§18.17.7.210), MIDB (§18.17.7.211), PHONETIC (§18.17.7.251), PROPER (§18.17.7.262), REPLACE (§18.17.7.272), REPLACEB (§18.17.7.273), REPT (§18.17.7.274), RIGHT (§18.17.7.275), RIGHTB (§18.17.7.276), SEARCH (§18.17.7.285), SEARCHB (§18.17.7.286), SUBSTITUTE (§18.17.7.304), T (§18.17.7.315), TEXT (§18.17.7.322), TRIM (§18.17.7.329), UPPER (§18.17.7.335), and VALUE (§18.17.7.337).

18.17.7.1 ABS

Syntax:

```
ABS ( x )
```

Description: Computes the absolute value of *x*.

Arguments:

Name	Type	Description
x	number	The value whose absolute value is to be determined.

Return Type and Value: number – The absolute value of *x*.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

ABS(10.5) results in 10.5
ABS(0) results in 0
ABS(-10.5) results in 10.5

end example]

18.17.7.2 ACCRINT

Syntax:

ACCRINT (*issue* , *first-interest* , *settlement* , *rate* , [*par*] , *frequency* [, [*basis*]])

Description: Computes the accrued interest for a security that pays periodic interest.

Mathematical Formula:

$$ACCRINT = par \times \frac{rate}{frequency} \times \sum_{i=1}^{NC} \frac{A_i}{NL_i}$$

where:

- A_iAi = number of accrued days for the i[th] quasi-coupon period within odd period.
- *frequency* = argument *frequency*
- *NC* = number of quasi-coupon periods that fit in odd period. If this number contains a fraction, raise it to the next whole number. The quasi-coupon period can be calculated in one of two following ways:
  - Odd long first coupon: by working backwards in time from the long coupon's interest payment date (first coupon date) and adding together the number of standard coupon periods that would fit in the long coupon, rounding up to the next whole number;
  - Odd long last coupon: by working forward in time from the long coupon's interest payment date (last coupon date before redemption) and adding together the number of standard coupon periods that would fit in the long coupon, rounding up to the next whole number.
- NL_iNLi = normal length in days of the i[th] quasi-coupon period within odd period.
- *par* = argument *par*
- *rate* = argument *rate*

Arguments:

Name	Type	Description
issue	number	The security's issue date.
first-interest	number	The security's first interest date.
settlement	number	The security's settlement date.
rate	number	The security's annual coupon rate.
par	number	The security's par value. If omitted, 1,000 is used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>frequency</td><td>number</td><td>The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. frequency is truncated to an integer.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr>
<tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr>
<tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr>
<tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, all dates with a day value of 31 are</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

Return Type and Value: number – The accrued interest for a security that pays periodic interest.

However, if

- *issue*, *first-interest*, or *settlement* is out of range for the current date system, `#NUM!` is returned
- *issue* ≥ *settlement*, `#NUM!` is returned
- *rate* or *par* ≤ 0, `#NUM!` is returned
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned
- *basis* < 0 or *basis* > 4, `#NUM!` is returned

[*Example*:

`ACCRINT(DATE(2006,3,1),DATE(2006,9,1),DATE(2006,5,1),0.1,1100,2,0)` results in `18.33`
`ACCRINT(DATE(2006,3,1),DATE(2006,9,1),DATE(2006,5,1),0.1,,2,0)` results in `16.67`

end example]

18.17.7.3 ACCRINTM

Syntax:

`ACCRINTM ( issue , settlement , rate , [ [ par ] [ , [ basis ]]] )`

Description: Computes the accrued interest for a security that pays interest at maturity.

Mathematical Formula:

$$ACCRINTM = par \times rate \times \frac{A}{D}$$

where:

- A = Number of accrued days counted according to a monthly basis. For interest at maturity items, the number of days from the issue date to the maturity date is used.

- D = Annual Year Basis.
- *par* = argument *par*
- *rate* = argument *rate*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Arguments:

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>issue</td><td>number</td><td>The security's issue date.</td></tr>
<tr><td>settlement</td><td>number</td><td>The security's settlement date.</td></tr>
<tr><td>rate</td><td>number</td><td>The security's annual coupon rate.</td></tr>
<tr><td>par</td><td>number</td><td>The security's par value. If omitted, 1,000 is used.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2 — Actual/360. Similar to Basis 1, but only has 360 days per year.
		3 — Actual/365. Similar to Basis 1, but always has 365 days per year.
		4 — European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The accrued interest for a security that pays interest at maturity.

However, if

- *issue* or *settlement* is out of range for the current date system, #NUM! is returned
- *issue* ≥ *settlement*, #NUM! is returned
- *rate* or *par* ≤ 0, #NUM! is returned
- *basis* < 0 or *basis* > 4, #NUM! is returned

[*Example*:

ACCRINTM(DATE(2006,3,1),DATE(2006,5,1),0.1,1100,0) results in 18.33333333
ACCRINTM(DATE(2006,3,1),DATE(2006,5,1),0.1,,0) results in 16.66666667
ACCRINTM(DATE(2006,3,1),DATE(2006,5,1),0.1,) results in 16.66666667

end example]

18.17.7.4 ACOS

Syntax:

ACOS (*x*)

Description: Computes the arc cosine of *x*.

Arguments:

Name	Type	Description
x	number	The value whose arc cosine is to be determined.

Return Type and Value: number – The arc cosine of *x*, in radians.

However, if *x* is outside the interval [-1,+1], #NUM! is returned

[*Example*:

ACOS(-1) results in 3.141592654
ACOS(0) results in 1.570796327
ACOS(1) results in 0

end example]

18.17.7.5 ACOSH

Syntax:

ACOSH (*x*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Computes the inverse hyperbolic cosine of *x*.

Arguments:

Name	Type	Description
x	number	The value whose inverse hyperbolic cosine is to be determined.

Return Type and Value: number – The inverse hyperbolic cosine of *x*.

However, if *x* < 1, `#NUM!` is returned.

[*Example*:

`ACOSH(1)` results in `0`
`ACOSH(10)` results in `2.993222846`
`ACOSH(100)` results in `5.298292366`

end example]

18.17.7.6 ADDRESS

Syntax:

```
ADDRESS ( row-number , col-number [ , [ ref-type ] [ , [ A1-ref-style-flag ]
    [ , sheet-name ] ] ] )
```

Description: Creates a cell address, given the specified row and column numbers.

Arguments:

Name	Type	Description
row-number	number	The number of the row.
col-number	number	The number of the column.
ref-type	number	The type of reference to return, as follows: 1 or omitted – Absolute row and column 2 – Absolute row; relative column 3 – Relative row; absolute column 4 – Relative row and column
A1-ref-style-	logical	The style of the reference. If TRUE or omitted, an A1-style

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
flag		reference (§18.17.2.3.1) is returned; otherwise, an R1C1-style reference (§18.17.2.3.2) is returned.
sheet-name	text	The name of the worksheet to be used. If omitted, no sheet name is used.

Return Type and Value: text – A cell address, given the specified row and column numbers.

However, if

- *row-number* or *col-number* < 1, `#NUM!` is returned.
- *ref-type* is outside the range 1–4, `#NUM!` is returned.

[*Example*:

In A1-reference style mode:

`ADDRESS(5,7,1)` results in `$G$5`
`ADDRESS(5,7,2)` results in `G$5`
`ADDRESS(5,7,3)` results in `$G5`
`ADDRESS(5,7,4)` results in `G5`
`ADDRESS(5,7,,,"Sheet1")` results in `Sheet1!$G$5`

In R1C1-reference style mode:

`ADDRESS(5,7,1,FALSE)` results in `R5C7`
`ADDRESS(5,7,2,FALSE)` results in `R5C[7]`
`ADDRESS(5,7,3,FALSE)` results in `R[5]C7`
`ADDRESS(5,7,4,FALSE)` results in `R[5]C[7]`

end example]

18.17.7.7 AMORDEGRC

Syntax:

AMORDEGRC (*cost* , *date-purchased* , *first-period* , *salvage* , *period* , *rate* [, [*basis*]])

Description: Computes the depreciation for each accounting period. (This function is provided for the French accounting system. If an asset is purchased in the middle of the accounting period, the prorated depreciation is taken into account. The function is similar to AMORLINC (§18.17.7.7), except that a depreciation coefficient is applied in the calculation depending on the life of the assets.)

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>cost</td><td>number</td><td>The cost of the asset.</td></tr>
<tr><td>date-purchased</td><td>number</td><td>The date of the purchase of the asset.</td></tr>
<tr><td>first-period</td><td>number</td><td>The date of the end of the first period.</td></tr>
<tr><td>salvage</td><td>number</td><td>The salvage value at the end of the life of the asset.</td></tr>
<tr><td>period</td><td>number</td><td>The period.</td></tr>
<tr><td>rate</td><td>number</td><td>The rate of depreciation.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr><tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The depreciation for each accounting period.

However, if

- *cost*, *salvage*, *period*, *or rate* < 0, `#NUM!` is returned.
- *date-purchased* or *first-period* is out of range for the current date system, `#NUM!` is returned.

 ©ISO/IEC 2016 – All rights reserved

- *basis* < 0 or *basis* > 4, `#NUM!` is returned.
- The life of the assets is between 0 and 1, 1 and 2, 2 and 3, or 4 and 5, `#NUM!` is returned.

This function returns the depreciation until the last period of the life of the assets or until the cumulated value of depreciation is greater than the cost of the assets minus the salvage value.

The depreciation coefficients are:

Life of assets (1/rate)	Depreciation Coefficient
Between 3 and 4 years	1.5
Between 5 and 6 years	2
More than 6 years	2.5

The depreciation rate grows to 50 percent for the period preceding the last period, and grows to 100 percent for the last period.

[*Example*:

`AMORDEGRC(2400,DATE(2008,8,19),DATE(2008,12,31),300,1,0.15,1)` results in `776.00`

end example]

18.17.7.8 AMORLINC

Syntax:

`AMORLINC (` *cost* `,` *date-purchased* `,` *first-period* `,` *salvage* `,` *period* `,` *rate* `[ , [` *basis* `] ] )`

Description: Computes the depreciation for each accounting period. (This function is provided for the French accounting system. If an asset is purchased in the middle of the accounting period, the prorated depreciation is taken into account.)

Arguments:

Name	Type	Description
cost	number	The cost of the asset.
date-purchased	number	The date of the purchase of the asset.
first-period	number	The date of the end of the first period.
salvage	number	The salvage value at the end of the life of the asset.
period	number	The period.
rate	number	The rate of depreciation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr>
<tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The depreciation for each accounting period.

However, if:

- *cost*, *salvage*, *period*, or *rate* < 0, `#NUM!` is returned.
- *date-purchased* or *first-period* is out of range for the current date system, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

©ISO/IEC 2016 – All rights reserved

[*Example*:

`AMORLINC(2400,DATE(2008,8,19),DATE(2008,12,31),300,1,0.15,1)` results in `360.00`

end example]

18.17.7.9 AND

Syntax:

AND (*argument-list*)

Description: Tests if all *arguments* in *argument-list* are TRUE. The function evaluates all arguments prior to returning a value.

Arguments:

Name	Type	Description
argument-list	logical, array, or cell reference	The *arguments* in *argument-list* designate the values to be tested. For an array or cell reference, a cell that contains text or is empty shall be ignored.

Return Type and Value: logical – TRUE if all *arguments* in *argument-list* are TRUE; otherwise, FALSE.

However, if no logical values are found, #VALUE! is returned.

[*Example*:

`AND(TRUE)` results in TRUE
`AND(TRUE,FALSE)` results in FALSE
`AND(10>5,3=1+2,5)` results in TRUE
`AND({10,5,6,7},TRUE,E6:F6)` results in TRUE, when E6 contains TRUE and F6 contains 10

end example]

18.17.7.10 AREAS

Syntax:

AREAS (*reference*)

Description: Finds the number of areas (§18.17.2.3) designated by *reference*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
reference	reference	A reference to a single cell or to a range of cells that can refer to multiple areas.

Return Type and Value: number – The number of areas designated by *reference*.

However, if the reference designates no areas, `#NUM!` is returned.

[*Example*:

`AREAS(E312)` results in `1`
`AREAS(E311:F313)` results in `1`
`AREAS((E312:F314,G316:H316,G311))` results in 3, given the union of the three areas
`AREAS((E312:F314 E313:F314 F312:F314))` results in 1, given the intersection of the three areas

end example]

18.17.7.11 ASC

Syntax:

ASC (*string*)

Description: For double-byte character set (DBCS) languages, converts all full-width (double-byte) characters to half-width (single-byte) characters.

Arguments:

Name	Type	Description
string	text	Designate the text to be converted. If *string* does not contain any full-width characters, nothing in *string* is converted.

Return Type and Value: text – The text resulting from the conversion.

[*Example*:

`ASC("ABC")` results in ABC
`ASC("エクセル")` results in ｴｸｾﾙ

end example]

18.17.7.12 ASIN

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
ASIN ( x )
```

Description: Computes the arc sine of *x*.

Arguments:

Name	Type	Description
x	number	The value whose arc sine is to be determined.

Return Type and Value: number – The arc sine of *x*, in radians.

However, if *x* is outside the interval [-1,+1], `#NUM!` is returned.

[*Example*:

`ASIN(-1)` results in `-1.570796327`
`ASIN(0)` results in `0`
`ASIN(1)` results in `1.570796327`

end example]

18.17.7.13 ASINH

Syntax:

```
ASINH ( x )
```

Description: Computes the inverse hyperbolic sine of *x*.

Arguments:

Name	Type	Description
x	number	The value whose inverse hyperbolic sine is to be determined.

Return Type and Value: number – The inverse hyperbolic sine of *x*.

[*Example*:

`ASINH(1)` results in `0.881373587`
`ASINH(10)` results in `2.99822295`
`ASINH(100)` results in `5.298342366`
`ASINH(0.5)` results in `0.481211825`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.14 ATAN

Syntax:

```
ATAN ( x )
```

Description: Computes the arc tangent of *x*.

Arguments:

Name	Type	Description
x	number	The value whose arc tangent is to be determined.

Return Type and Value: number – The arc tangent of *x*, in radians.

[*Example*:

`ATAN(-1)` results in `-0.785398163`
`ATAN(0)` results in `0`
`ATAN(1)` results in `0.785398163`
`ATAN(-10)` results in `1.471127674`
`ATAN(10)` results in `1.471127674`

end example]

18.17.7.15 ATAN2

Syntax:

```
ATAN2 ( x , y )
```

Description: Computes the arc tangent of the coordinates *x* and *y*.

Arguments:

Name	Type	Description
x	number	The first coordinate.
y	number	The second coordinate.

Return Type and Value: number – The arc tangent of *y*/*x*, in radians.

However, if both *x* and *y* are zero, `#DIV/0!` is returned.

[*Example*:

`ATAN2(1,1)` results in `0.785398163`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ATAN2(-2,2) results in 2.35619449
ATAN2(3,-3) results in -0.785398163

end example]

18.17.7.16 ATANH

Syntax:

ATANH (*x*)

Description: Computes the inverse hyperbolic tangent of *x*.

Arguments:

Name	Type	Description
x	number	The value whose inverse hyperbolic tangent is to be determined.

Return Type and Value: number – The inverse hyperbolic tangent of *x*.

However, if *x* is outside the interval [-1,+1], #NUM! is returned.

[*Example*:

ATANH(-0.999999) results in -7.254328619
ATANH(0) results in 0
ATANH(0.999999) results in 7.254328619

end example]

18.17.7.17 AVEDEV

Syntax:

AVEDEV (*argument-list*)

Description: Computes the average of the absolute deviations of a set of data points from their mean. AVEDEV is a measure of the variability in a data set.

Mathematical Formula:

The average of the absolute deviations of a set of data points from their mean is as follows:

$$\frac{1}{n}\sum|x - \bar{x}|$$

where:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- *n* = the number of *arguments* in *argument-list*
- *x* = an *argument* in *argument-list*
- $\bar{x}$= the *x* mean value

Arguments:

Name	Type	Description
argument-list	logical, number, name, array, or reference that contains a number. The list can be a single argument that is an array or a reference to an array.	The *arguments* in *argument-list* designate the values for which the average of the absolute deviations is to be computed. Logical values and text representations of numbers occurring directly in the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The average of the absolute deviations of a set of data points from their mean.

[*Example*:

AVEDEV(-3.5,1.4,6.9,-4.5) results in 4.075
AVEDEV({-3.5,1.4,6.9,-4.5}) results in 4.075

end example]

18.17.7.18 AVERAGE

Syntax:

AVERAGE (*argument-list*)

Description: Computes the arithmetic mean of the numeric values of its arguments.

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, or reference that contains a number.	The *arguments* in *argument-list* designate the values to be averaged. An argument that is a logical value or the text representation of a number shall be counted. If an array or cell reference argument contains logical values, text, or empty cells, those values shall be ignored; however, cells having the value 0value 0 shall be counted. [*Note*: The function AVERAGEA (§18.17.7.18) does include cell

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		reference arguments that refer to logical values or text representations of numbers. *end note*]

Return Type and Value: number – The arithmetic mean of the values of its arguments.

[*Example*:

`AVERAGE(1,2,3,4,5)` results in 3
`AVERAGE({1,2;3,4})` results in `2.5`
`AVERAGE({1,2,3,4,5},6,"7")` results in 4
`AVERAGE({1,"2",TRUE,4})` results in `2.5`, as the logical value and numeric text are ignored

end example]

18.17.7.19 AVERAGEA

Syntax:

AVERAGEA (*argument-list*)

Description: Computes the arithmetic mean of the values of its arguments.

Arguments:

Name	Type	Description
argument-list	logical, number, text, or reference that contains a number.	The *arguments* in *argument-list* designate the values to be averaged. An argument that is a logical value or the text representation of a number shall be counted. Arguments with value TRUE evaluate to 1; arguments with value FALSE evaluate to 0. An array or cell reference argument that contains text evaluates to 0. If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored.

[*Note*: The function AVERAGE (§18.17.7.18) does not include cell reference arguments that refer to logical values or text representations of numbers. *end note*]

Return Type and Value: number – The arithmetic mean of the values of its arguments.

[*Example*:

`AVERAGEA(10,E1)`, where E1 is an empty cell, results in `10`, as E1 is ignored
`AVERAGEA(10,E2)`, where E2 contains TRUE, results in `5.5`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

AVERAGEA(10,E3), where E3 contains FALSE, results in 5

end example]

18.17.7.20 AVERAGEIF

Syntax:

AVERAGEIF (*cell-range* , *selection-criteria* [, *average-range*])

Description: Applies selection criteria on the values in one range of cells and averages the values of the cells in a corresponding range.

Arguments:

Name	Type	Description
cell-range	reference	The range of cells to be inspected. Cells in *cell-range* that contain TRUE or FALSE are ignored. If a cell is an empty cell, it is ignored.
selection-criteria	number, expression, reference, text	Designates the cells that are to be averaged. In the case of text, *selection-criteria* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. *selection-criteria* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~). If a cell in *selection-criteria* is empty, it is treated as if it contained 0.
average-range	reference	Designates the cells whose values are averaged. In this case, *average-range* need not have the same size and shape as *cell-range*. The actual cells that are averaged are determined by using the top, left cell in *average-range* as the beginning cell, and then including cells that correspond in size and shape to *cell-range*. If *average-range* is omitted, *cell-range* also designates the cells whose values are averaged. If a cell is an empty cell, it is ignored.

Return Type and Value: number – The average of the values of the cells corresponding to those selected.

However, if no cells in the range meet the criteria, the return value is unspecified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Assuming A2:A4 contains 10, 20, and 30:

AVERAGEIF(A2:A4,“>15”) results in 25, the average of 20 and 30.

end example]

18.17.7.21 AVERAGEIFS

Syntax:

AVERAGEIFS (*average-range* , *cell-range-1* , *selection-criteria-1*
[, *cell-range-2* , *selection-criteria-2* [, …]])

Description: The average of the values of all cells that meet multiple criteria.

Arguments:

Name	Type	Description
average-range	reference	Designates the cells whose values are averaged. In this case, *average-range* need not have the same size and shape as *cell-range-1* through *cell-range-n*. The actual cells that are added are determined by using the top, left cell in *average-range* as the beginning cell, and then including cells that correspond in size and shape to *cell-range-1* through *cell-range-n*. If a cell in *average-range* is empty, that cell is ignored. Each cell in *average-range* is used in the average calculation only if all of the corresponding criteria specified are true for that cell.
cell-range-1	number, expression, reference, text	Designates the first range of cells to be inspected.
selection-criteria-1	reference, text	*selection-criteria-1* specifies the criteria for the first range of cells that is averaged. In the case of text, *selection-criteria-1* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. If a cell in any selection criteria range is empty, it is treated as if its value was 0. Cells that contain TRUE evaluate to 1; cells in any range that contain FALSE evaluate to 0. *selection-criteria-1* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~).
cell-range-n	number, expression,	The optional arguments *selection-criteria-2* through *selection-criteria-n* have corresponding arguments *cell-*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
	reference, text	*range-2* through *cell-range-n*, and have the same semantics as *selection-criteria-1* and *cell-range-1*, respectively.
selection-criteria-n	reference, text	

Return Type and Value: number – The average of the cells corresponding to those selected.

However, if

- Cells in *average-range* are empty or contain text values that cannot be translated into numbers, the return value is unspecified.
- There are no cells that meet all the criteria, the return value is unspecified.

[*Example*: Given the following data:

	A	B	C	D
1	Student	First Quiz Grade	Second Quiz Grade	Final Exam Grade
2	Emilio	75	85	87
3	Julie	94	80	88
4	Hans	86	93	Incomplete
5	Frederique	Incomplete	75	75

`AVERAGEIFS(B2:B5,B2:B5,">70",B2:B5,"<90")` results in `80.5` (the average for all students all first quiz grades that are between 70 and 90)

`AVERAGEIFS(D2:D5,D2:D5,"<>Incomplete",D2:D5,">80")` results in `87.5` (the average for all students all first quiz grades that are above 80 and not marked "Incomplete")

`AVERAGEIFS(B2:D5,B2:B5,"<>Incomplete",C2:C5,"<>Incomplete",D2:D5, "<>Incomplete")` results in 82.375 (the average grades for all students who do not have incomplete grades)

end example]

18.17.7.22 BAHTTEXT

Syntax:

BAHTTEXT (*number*)

Description: Produces a string containing *number* formatted according to the Thai convention.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Name	Type	Description
number	number	The value to be formatted.

Return Type and Value: text – The text containing *number* formatted.

[*Example*:

BAHTTEXT(1234) results in **หนึ่งพันสองร้อยสามสิบสี่บาทถ้วน**

end example]

18.17.7.23 BESSELI

Syntax:

```
BESSELI ( x , n )
```

Description: The modified Bessel function In(x), which is equivalent to the Bessel function Jn(x) evaluated for purely imaginary arguments.

Mathematical Formula:

The n-th order modified Bessel function of the variable x is:

$$\mathrm{I_n}(x) = (i)^{-n} J_n\,(ix)$$

where:

- x = argument x
- n = argument n

Arguments:

Name	Type	Description
x	number	The value at which to evaluate the function.
n	number	The order of the Bessel function. This value is truncated to an integer.

Return Type and Value: number – The Bessel function In(x).

However, if $n < 0$, #NUM! is returned.

[*Example*:

BESSELI(-5.6,0) results in 46.73755194

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`BESSELI(2.345,5)` results in `0.023137792`

end example]

18.17.7.24 BESSELJ

Syntax:

`BESSELJ ( x , n )`

Description: The Bessel function Jn(x).

Mathematical Formula:

The n-th order Bessel function of the variable x is:

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\,\Gamma(n+k+1)} \left(\frac{x}{2}\right)^{n+2k}$$

where:

$$\Gamma(n+k+1) = \int_0^{\infty} e^{-x} x^{n+k}\, dx$$

is the Gamma function, and

- x = argument x
- n = argument n

Arguments:

Name	Type	Description
x	number	The value at which to evaluate the function.
n	number	The order of the Bessel function. This value is truncated to an integer.

Return Type and Value: number – The Bessel function Jn(x).

However, if $n < 0$, `#NUM!` is returned.

[*Example*:

`BESSELJ(-5.6,0)` results in `0.026970887`
`BESSELJ(2.345,5)` results in `0.014627862`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.25 BESSELK

Syntax:

```
BESSELK ( x , n )
```

Description: The modified Bessel function Kn(x), which is equivalent to using the Bessel function Jn(x) and Yn(x).

Mathematical Formula:

The n-th order modified Bessel function of the variable x is:

$$K_n(x) = \frac{p}{2} i^{n+1}[J_n(ix) + iY_n(ix)]$$

where:

- J_n is the J Bessel function
- n = argument n
- p = π
- x = argument x
- Y_nYn is the Y Bessel function
- **Arguments:**

Name	Type	Description
x	number	The value at which to evaluate the function.
n	number	The order of the Bessel function. This value is truncated to an integer.

Return Type and Value: number – The Bessel function Kn(x).

However, if $n < 0$, #NUM! is returned.

[*Example*:

`BESSELK(2.345,5)` results in `3.904137225`

end example]

18.17.7.26 BESSELY

Syntax:

```
BESSELY ( x , n )
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Weber's Bessel function Yn(x).

Mathematical Formula:

The n-th order Bessel function of the variable x is:

$$Y_n(x) = \lim_{v \to n} \frac{J_v(x)\cos(v\,\pi) - J_{-v}(x)}{\sin(v\,\pi)}$$

where:

- n = argument n
- x = argument x

Arguments:

Name	Type	Description
x	number	The value at which to evaluate the function.
n	number	The order of the Bessel function. This value is truncated to an integer.

Return Type and Value: number – The Weber's Bessel function Yn(x).

However, if $n < 0$, `#NUM!` is returned.

[*Example*:

`BESSELY(2.345,5)` results in `-4.98977884`

end example]

18.17.7.27 BETADIST

Syntax:

`BETADIST ( x , alpha , beta [ , [ A ] , [ B ]] )`

Description: Computes the cumulative beta probability density function.

Arguments:

Name	Type	Description
x	number	The value between A and B at which to evaluate the function.
alpha	number	A parameter of the distribution.
beta	number	A parameter of the distribution.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
A	number	The lower bound to the interval of *x*. If omitted, the lower bound is 0.
B	number	The upper bound to the interval of *x*. If omitted, the upper bound is 1.

Return Type and Value: number – The cumulative beta probability density function.

However, if

- *alpha* or *beta* ≤ 0, `#NUM!` is returned.
- *x* < *A*, *x* > *B*, or *A* = *B*, `#NUM!` is returned.

[*Example*:

`BETADIST(0.5,1,2)` results in `0.75`
`BETADIST(0.5,1,2,-4.5,7.3)` results in `0.66791152`
`BETADIST(0.5,1,2,,2.3)` results in `0.387523629`

end example]

18.17.7.28 BETAINV

Syntax:

`BETAINV ( probability , alpha , beta [ , [ A ] , [ B ]] )`

Description: Computes the inverse of the cumulative distribution function for a specified beta distribution. Given a value for *probability*, `BETAINV` is used to seek for the value *x* such that `BETADIST(`*x, alpha, beta, A, B*`)` = *probability*. Thus, precision of `BETAINV` depends on precision of `BETADIST`.

Arguments:

Name	Type	Description
probability	number	A probability associated with the beta distribution.
alpha	number	A parameter of the distribution.
beta	number	A parameter of the distribution.
A	number	The lower bound to the interval of *x*. If omitted, the lower bound is 0.
B	number	The upper bound to the interval of *x*. If omitted, the upper bound is 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The inverse of the cumulative distribution function for a specified beta distribution.

However, if

- *alpha* or *beta* ≤ 0, `#NUM!` is returned.
- *probability* ≤ 0 or *probability* ≥ 1, `#NUM!` is returned.
- $A \geq B$, `#NUM!` is returned.
- The search has not converged after some implementation-defined number of iterations, `#N/A` is returned.

[*Example*:

`BETAINV(0.5,1,2)` results in `0.29289341`
`BETAINV(0.5,1,2,-4.5,7.3)` results in `-1.043857765`
`BETAINV(0.5,1,2,,2.3)` results in `0.673654842`

end example]

18.17.7.29 BIN2DEC

Syntax:

`BIN2DEC ( number )`

Description: Makes the decimal equivalent of *number*.

Arguments:

Name	Type	Description
number	number	A 10-digit binary number that is to be converted to a decimal number. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (10th bit from the right) representing the sign bit.

Return Type and Value: number – The decimal equivalent of *number*.

However, if

- *number* contains one or more non-binary digits, `#NUM!` is returned.
- *number* contains more than 10 binary digits; that is, *number* is outside the range `1000000000` (-512 decimal) to `0111111111` (511 decimal), inclusive, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

`BIN2DEC(111)` results in 7
`BIN2DEC(11111111)` results in 255
`BIN2DEC(1111111110)` results in -2
`BIN2DEC(1000000000)` results in -512

end example]

18.17.7.30 BIN2HEX

Syntax:

`BIN2HEX ( number [ , num-hex-digits ] )`

Description: Makes the uppercase hexadecimal equivalent of *number*, with the result having *num-hex-digits* digits.

Arguments:

Name	Type	Description
number	number	A 10-digit binary number that is to be converted to a hexadecimal string. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (10th bit from the right) representing the sign bit.
num-hex-digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-hex-digits* is ignored and the result has 10 digits. If *num-hex-digits* is omitted, the minimum number of digits is used in the result. *num-hex-digits* is truncated to an integer.

Return Type and Value: text – The uppercase hexadecimal equivalent of *number*.

However, if

- *number* contains one or more non-binary digits, `#NUM!` is returned.
- *number* contains more than 10 binary digits; that is, *number* is outside the range `1000000000` (200 hex, -512 decimal) to `0111111111` (1FF hex, 511 decimal), inclusive, `#NUM!` is returned.
- *number* needs more digits that *num-hex-digits*, `#NUM!` is returned.
- *num-hex-digits* ≤ 0 or > 10, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`BIN2HEX(1)` results in `1`
`BIN2HEX(1,4)` results in `0001`
`BIN2HEX(111111)` results in `3F`
`BIN2HEX(1111000000)` results in `FFFFFFFFC0`
`BIN2HEX(1000000000,3)` results in `FFFFFFFE00`

end example]

18.17.7.31 BIN2OCT

Syntax:

`BIN2OCT ( number [ , num-oct-digits ] )`

Description: Makes the octal equivalent of *number*, with the result having *num-oct-digits* digits.

Arguments:

Name	Type	Description
number	number	A 10-digit binary number that is to be converted to an octal string. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (10th bit from the right) representing the sign bit.
num-oct-digits	number	*num-oct-digits* is the number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-oct-digits* is ignored and the result has 10 digits. If *num-oct-digits* is omitted, the minimum number of digits is used in the result. *num-oct-digits* is truncated to an integer.

Return Type and Value: text – The octal equivalent of *number*.

However, if

- *number* contains one or more non-binary digits, `#NUM!` is returned.
- *number* contains more than 10 binary digits; that is, *number* is outside the range `1000000000` (1000 octal, -512 decimal) to `0111111111` (0777 octal, 511 decimal), inclusive, `#NUM!` is returned.
- *number* needs more digits that *num-oct-digits*, `#NUM!` is returned.
- *num-oct-digits* < 0 or > 10, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

`BIN2OCT(1)` results in `1`
`BIN2OCT(1,4)` results in `0001`
`BIN2OCT(111111)` results in `77`
`BIN2OCT(1111000000)` results in `7777777700`
`BIN2OCT(1000000000,3)` results in `7777777000`

end example]

18.17.7.32 BINOMDIST

Syntax:

`BINOMDIST` (*number-successes* , *number-trials* , *success-probability* , *cumulative-flag*)

Description: Computes the individual term binomial distribution probability.

Mathematical Formula:

The binomial probability mass function is:

$$b(x,n,p) = \binom{n}{x} p^x (1-p)^{n-x}$$

where:

$$\binom{n}{x}$$

is `COMBIN(n,x)`.

The cumulative binomial distribution is:

$$B(x,n,p) = \sum_{y=0}^{x} b(y,n,p)$$

where:

- n = argument *number-trials*
- p = argument *success-probability*
- x = argument *number-successes*

Arguments:

Name	Type	Description
number-successes	number	The number of successes in *number-trials*, truncated to an integer.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
number-trials	number	The number of independent trials, truncated to an integer.
success-probability	number	The probability of success on each trial.
cumulative-flag	logical	Determines the form of the function. If TRUE, then the cumulative distribution function is returned, which is the probability that there are at most *number-successes* successes; if FALSE, the probability mass function is returned, which is the probability that there are *number-successes* successes.

Return Type and Value: number – The individual term binomial distribution probability.

However, if

- *number-successes* < 0 or *number-successes* > *number-trials*, #NUM! is returned.
- *success-probability* < 0 or *success-probability* > 1, #NUM! is returned.

[*Example*:

`BINOMDIST(6,10,0.5,FALSE)` results in `0.205078125`
`BINOMDIST(6,10,0.5,TRUE)` results in `0.828125`

end example]

18.17.7.33 CEILING

Syntax:

`CEILING ( x , significance )`

Description: Computes a value that is *x* rounded-up, away from zero, to the nearest multiple of *significance*. Regardless of the sign of *x*, a value is rounded up when adjusted away from zero.

Arguments:

Name	Type	Description
x	number	The value to be rounded
significance	number	The multiple to which *x* is to be rounded. If *x* is negative, and *significance* is negative, then the value is rounded down (away from zero). If *x* is negative, and *significance* is positive, then the value is rounded up, towards zero.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The rounded-up value of *x*.

However, if *x* and *significance* have different signs, `#NUM!` is returned.

[*Example*:

`CEILING(2.5,1)` rounds 2.5 up to nearest multiple of 1; that is, to 3
`CEILING(-2.5,-2)` rounds -2.5 up to nearest multiple of -2; that is, to `-4`
`CEILING(1.5,0.1)` rounds 1.5 up to the nearest multiple of 0.1; that is, to `1.5`
`CEILING(0.234,0.01)` rounds 0.234 up to the nearest multiple of 0.01; that is, to `0.24`

end example]

18.17.7.34 CELL

Syntax:

`CELL ( ` *category* `[ , ` *reference* ` ] )`

Description: Retrieves information about the formatting, location, or contents of the upper-left cell indicated by *reference*. *category* indicates the kind of information to be retrieved.

Arguments:

Name	Type	Description
category	text	The category string as defined in the table following.
reference	reference	Refers to the cell whose category information is being requested. If *reference* is a cell range, the first cell in that range is the cell whose category information is being requested. If *reference* is omitted, the information retrieved pertains to the most recent cell whose value was changed. For the category `"format"`, if *reference* designates a cell formatted with a built-in number format, the number format string is as defined in the table following.

category	Meaning	Result Type
`"address"`	Reference of the first cell in *reference*.	text
`"col"`	Column number of the cell in *reference*.	number
`"color"`	1 if the cell is formatted in color for negative values; otherwise, 0. 0 if the cell does not contain a number.	number
`"contents"`	Value of the upper-left cell in *reference*.	Text or number

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

category	Meaning	Result Type
`"filename"`	Fully qualified filename of the file that contains *reference*. However, if the worksheet that contains *reference* has not yet been saved, the filename is an empty string.	text
`"format"`	Number format of the cell. (See the discussion of formats below.) The number format string has `"-"` appended if the cell is formatted in color for negative values. The number format string has `"()"` appended if the cell is formatted in color for positive or all values.	text
`"parentheses"`	1 if the cell is formatted with parentheses for positive or all values; otherwise, 0. 0 if the cell does not contain a number.	number
`"prefix"`	Text value corresponding to the label prefix of the cell, as follows: • Single quotation mark (') if the cell contains left-aligned text • Double quotation mark (") if the cell contains right-aligned text • Caret (^) if the cell contains centered text • Backslash (\) if the cell contains fill-aligned text • Empty string if the cell contains anything else	text
`"protect"`	0 if the cell is not locked; otherwise, 1.	number
`"row"`	Row number of the cell in reference.	number
`"type"`	Text value corresponding to the type of data in the cell. • `"b"` (blank) if the cell is empty • `"l"` (label) if the cell contains a text constant • `"v"` (value) if the cell contains anything else	text
`"width"`	Column width of the cell rounded off to an integer. Each unit of column width is equal to the width of one character in the default font size.	number

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If the SpreadsheetML is intended to be used in certain non-English locales, the *category* string can be the English value shown in the table above, or the translation shown in the following table. Locales not specified in the table below shall only use the English versions of the *category* string.

[*Note*: Using translated versions of the *category* string is strongly discouraged, as spreadsheet applications might not support these translations. *end note*]

Locale	address	col	color	contents	filename	format
az-latn-az	ünvan	col	rəng	mündəricat	fayladi	format
ca-es	dirección	columna	color	contenido	nombrearchivo	formato
cs-cz	adresa	sloupec	barva	obsah	názevsouboru	formát

©ISO/IEC 2016 – All rights reserved

Locale	address	col	color	contents	filename	format
da-dk	adresse	kolonne	farve	indhold	filnavn	format
de-de	adresse	spalte	farbe	inhalt	dateiname	format
es-es	direccion	columna	color	contenido	nombrearchivo	formato
et-ee	aadress	veerg	värv	sisukord	failinimi	vorming
fi-fi	osoite	sarake	väri	sisältö	tiedostonnimi	muoto
fr-fr	adresse	colonne	couleur	contenu	nomfichier	format
hu-hu	cím	oszlop	szín	tartalom	filenév	forma
it-it	indirizzo	col	colore	contenuto	nomefile	formato
kk-kz	мекенжай	баған	түс	мазмұны	файлатауы	пішім
lv-lv	adrese	kolonna	krāsa	saturs	faila_nosaukums	formāts
nb-no	adresse	kol	farge	innhold	filnavn	format
nl-nl	adres	kolom	kleur	inhoud	bestandsnaam	notatie
pl-pl	adres	kolumna	kolor	zawartość	nazwa_pliku	format
pt-br	endereço	col	cor	conteúdo	nome.arquivo	formato
pt-pt	endereço	col	cor	conteúdo	nome.ficheiro	formato
ru-ru	адрес	столбец	цвет	содержимое	имяфайла	формат
sk-sk	adresa	stĺpec	farba	obsah	názovsúboru	formát
sl-si	address	sto	color	contents	filename	format
sv-se	adress	kol	färg	innehåll	filnamn	format
tr-tr	adres	süt	renk	içerik	dosyaadi	biçim
uk-ua	адреса	стовпець	колір	вміст	ім`я_файлу	формат

Locale	parentheses	prefix	protect	row	type	width
az-latn-az	parentheses	prefix	protect	sətir	tip	en
ca-es	parentesis	prefijo	proteger	fila	tipo	ancho
cs-cz	závorky	prefix	zámek	řádek	typ	šířka
da-dk	parenteser	foranstillet	beskyt	række	værditype	bredde
de-de	klammern	präfix	schutz	zeile	typ	breite
es-es	parentesis	prefijo	proteger	fila	tipo	ancho
et-ee	sulud	eesliide	kaitse	rida	tüüp	laius
fi-fi	sulkeet	etuliite	suojaus	rivi	tyyppi	leveys

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Locale	parentheses	prefix	protect	row	type	width
fr-fr	parentheses	prefixe	protege	ligne	type	largeur
hu-hu	zárójelek	előtag	védett	sor	típus	széles
it-it	parentesi	prefisso	proteggi	riga	tipo	larghezza
kk-kz	жақшалар	префикс	қорғаныс	жол	түр	ені
lv-lv	iekavas	prefikss	aizsargāt	rinda	tips	platums
nb-no	parenteser	prefiks	beskytt	rad	verditype	bredde
nl-nl	haakjes	voorvoegsel	bescherming	rij	type	breedte
pl-pl	nawiasy	prefiks	ochrona	wiersz	typ	szerokość
pt-br	parênteses	prefixo	proteger	lin	tipo	largura
pt-pt	parênteses	prefixo	proteger	lin	tipo	largura
ru-ru	скобки	префикс	защита	строка	тип	ширина
sk-sk	zátvorky	vloženýznak	chrániť	riadok	typ	šírka
sl-si	parentheses	prefix	protect	vrstica	type	širina
sv-se	parenteser	prefix	skydd	rad	typ	bredd
tr-tr	ayraç	önek	koruma	sat	tür	genişlik
uk-ua	дужки	префікс	захист	рядок	тип	довжина

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: various (see table above) – The value corresponding to *category*, and whose type is shown in the category value table above.

When the category parameter is `"format"`, then the value returned depends upon the number format of the upper-left cell of *reference*, and, more specifically, upon the number format code of the upper-left cell of *reference*. Depending upon the number format code of the appropriate cell, the result value of CELL when the category is `"format"` is based upon the rules defined below.

First, some observations regarding the rules are in order:

- The various "sections" of the number format code are referred to within the rules. For more information on sections in the number format code, see numFmts (Number Formats) (§18.8.31).
- There are cases in which it is useful to discuss the characters from the number format code that are dependent upon the value in the cell. Instead of representing text or spacing in the cell's display text, these characters interpret, in some fashion, the value to be displayed. In the rules, these characters are referred to as "interpreted characters" of the number format code. The following table shows all the interpreted characters:

©ISO/IEC 2016 – All rights reserved

Interpreted Characters
0
#
@
d
m
y
h
s
?
AM/PM
A/P
g
e
r

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- There are cases in which it is also useful to discuss runs of similar interpreted characters. [*Example*: Each "d", "m", and "y" within the number format code "dd/mm/yyyy" does not represent a separate interpretation of the day of the date value to be represented, and instead helps to make up a representation of the day of the date that is two digits in length. *end example*] These runs of similar characters are referred to below as "interpreted symbols" since multiple characters are used, but the result is a single symbolic representation of at least part of the value.
- Since there are multiple different symbols for days, months, years, hours, minutes, seconds, and AM/PM, it is sometimes useful to discuss all of the representations of each of these. When any of "Day", "Month", "Year", "Hour", "Minute", etc., is referred to within a rule, what is meant is any of the possible representations for that date/time portion. For example, a "Day" symbol would be any of d, dd, ddd, or dddd.
- The use of characters such as "0", "E", and "%" as a symbol within a number format code does not include usages of these same characters either escaped (by preceding them with a backslash character "\") or as a part of a quoted string.

Here are the rules to determine the result value:

1. If the first interpreted symbols within the first section are any of the date or time characters (any of "y", "m", "d", "h", "m", "s") then the first one or two characters of the return value are determined by the order of interpreted symbols (including any interpreted symbols, not just date/time symbols) according

 ©ISO/IEC 2016 – All rights reserved

to the following table. ("anything" can mean "no additional symbols". If the type is "anything except <type>", then the excepted symbol type cannot follow the previously specified symbol.)

Interpreted Symbols in Order	Return Value Characters
Day, month, year, anything	"D1"
Day, month, anything besides year, anything	"D2"
Month, year, anything	"D3"
Month, day, year, anything	"D4"
Month, day, anything besides year, anything	"D5"
Hours, minutes, seconds, AM/PM, anything	"D6"
Hours, minutes, AM/PM, anything	"D7"
Hours, minutes, seconds, anything besides AM/PM, anything	"D8"
Hours, minutes, anything besides seconds or AM/PM, anything	"D9"
Any other combination of symbols	"G" or "C" depending on whether there is a $ in the first condition

2. Otherwise, the first character of the return value is determined based upon the referenced cell's number format code according to the rules in the table below:

First Section of the Number Format Code		First Return Value Character
Absent Characters (as Symbols)	Present Characters (as Symbols)	
	@	"G"
@	$	"C"
@, $	%	"P"
@, $, %	E	"S"
@, $, %, E	A run of any combination of one or more "0", "#", and "?" characters, followed by a comma, followed by a run of any combination of one or more "0", "#", and "?" characters.	","

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

First Section of the Number Format Code		First Return
	This run cannot be preceded by a period (".") or by any interpreted characters.	
@, $, %, E, or a run of characters containing a comma as described in the row above	At least one of "0", "?", or "#"	"F"
Any of the above conditions	Anything	"G"

3. The final value is determined by appending any of the applicable characters from the table below to the return value characters obtained from the previous two tables:

Case	Characters to Append
The first character of the return value is C, F, S, P, or "," and the number format code contains any of "?", "0", or "#" as a symbol.	The decimal number equal to the total number of "#", "?", and "0" characters to the right of the first "." within the first section
The first character of the return value was C, and the number format code does not contains any of "?", "0", or "#" as a symbol.	15 or the length of the string immediately following the first "$" sign in the number format code that is a symbol.
The first section of the number format code contains an open parenthesis ("(") as a symbol.	"()"
The second section of the number format code contains [Red], [Black], [Green], [White], [Blue], [Magenta], [Yellow], or [Cyan] as a symbol.	"-"

However, if *category* is not one of the defined values, #VALUE ! is returned.

[*Example*:

`CELL("address",A10)` might result in `$E$289`
`CELL("contents",A10:B10)`, results in `xxx`, when A10 contains `xxx`, and B10 contains anything
`CELL("filename",A10)` might result in `E:\Formulas\[Test.xlsx]Sheet1`
`CELL("format",A10)` results in `G`, when A10 contains `xxx`
`CELL("format",A10)` results in `F2-`, when A10 contains `(123.00)`
`CELL("format",A10)` results in `C3-`, when A10 contains `$123,456.780`
`CELL("format",A10)` results in `S3`, when A10 contains `1.235E+05`
`CELL("prefix",A10)` results in `'`, when A10 contains `xxx`
`CELL("type",A10)` results in `l`, when A10 contains `xxx`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.35 CHAR

Syntax:

CHAR (*x*)

Description: Determines the character that is represented by the value *number*. On the Macintosh platform, the Macintosh character set is used. On all other platforms, the Latin character set with IANA name `iso-8859-1` is used.

Arguments:

Name	Type	Description
x	number	A value in the range 1–255, which designates the character.

Return Type and Value: text – The character represented by the value *number*.

[*Example*:

CHAR(65) results in A
CHAR(A10) results in A, when A10 contains 65

end example]

18.17.7.36 CHIDIST

Syntax:

CHIDIST (*x* , *degrees-freedom*)

Description: Computes the one-tailed probability of the chi-squared distribution.

Mathematical Formula:

$CHIDIST = P(X > x)$

where:

- X = an χ2 random variable
- x = argument x

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
x	number	The value at which the distribution is to be evaluated.
degrees-freedom	number	The number of degrees of freedom, truncated to an integer.

Return Type and Value: number – The one-tailed probability of the chi-squared distribution.

However, if

- *x* < 0, `#NUM!` is returned.
- *degrees-freedom* < 1 or *degrees-freedom* > 10^{10}, `#NUM!` is returned.

[*Example*:

`CHIDIST(3.5,4)` results in `0.47787835`
`CHIDIST(12.34,7)` results in `0.089917721`

end example]

18.17.7.37 CHIINV

Syntax:

`CHIINV (` *probability* `,` *degrees-freedom* `)`

Description: Computes the inverse of the one-tailed probability of the chi-squared distribution. Given a value for *probability*, `CHIINV` seeks for a value *x* such that `CHIDIST(`*x*`,` *degrees-freedom*`)` = *probability*. Thus, precision of `CHIINV` depends on precision of `CHIDIST`.

Arguments:

Name	Type	Description
probability	number	A probability associated with the chi-squared distribution.
degrees-freedom	number	The number of degrees of freedom, truncated to an integer.

Return Type and Value: number – The inverse of the one-tailed probability of the chi-squared distribution.

However, if

- *probability* < 0 or *probability* > 1, `#NUM!` is returned.
- *degrees-freedom* < 1 or *degrees-freedom* ≥ 10^{10}, `#NUM!` is returned.
- the implementation determines that a return value cannot be computed, `#N/A` is returned

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`CHIINV(0.5,4)` results in `3.356694001`
`CHIINV(0.3,7)` results in `8.38343064`

end example]

18.17.7.38 CHITEST

Syntax:

`CHITEST` (*actual-range* , *expected-range*)

Description: Computes the test for independence. `CHITEST` returns the value from the chi-squared distribution for the statistic and the appropriate degrees of freedom.

Mathematical Formula:

The χ2 test first calculates a χ2 statistic using the formula:

$$x^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where:

- A_{ij} = actual frequency in the i-th row, j-th column of the argument *actual-range*
- c = number of columns in argument *actual-range* (or argument *expected-range*)
- E_{ij} = expected frequency in the i-th row, j-th column of the argument *expected-range*
- r = number or rows in argument *actual-range* (or argument *expected-range*)

CHITEST uses the χ2 distribution with an appropriate number of degrees of freedom, df. If r > 1 and c > 1, then df = (r - 1)(c - 1). If r = 1 and c > 1, then df = c - 1 or if r > 1 and c = 1, then df = r - 1.

Arguments:

Name	Type	Description
actual-range	reference	The range of data that contains observations to test against expected values.
expected-range	reference	The range of data that contains the ratio of the product of row totals and column totals to the grand total.

Return Type and Value: number – The value from the chi-squared distribution for the statistic and the appropriate degrees of freedom.

However, if:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- The number of rows and columns is exactly one, the return value is unspecified.
- *actual-range* and *expected-range* have a different number of data points, #N/A is returned.

[*Example*: Given the following data:

	A	B	C
1	Men (Actual)	Women (Actual)	Description
2	58	35	Agree
3	11	25	Neutral
4	10	23	Disagree
5	Men (Expected)	Women (Expected)	Description
6	45.35	47.65	Agree
7	17.56	18.44	Neutral
8	16.09	16.91	Disagree

`CHITEST(A2:B4,A6:B8)` results in `0.000308`

end example]

18.17.7.39 CHOOSE

Syntax:

CHOOSE (*index* , *argument-list*)

Description: Selects the *argument* in *argument-list* that corresponds by position to *index*.

Arguments:

Name	Type	Description
index	number	An index into *argument-list*, truncated to an integer. The value of *index* shall be in the position range 1–*n*, where *argument-1* is position 1, *argument-2* is position 2, and so on up to *argument-n*. If *index* is an array, the value or every element in that array is evaluated, and if the formula is an array formula, the result is an array of chosen values.
argument-list	any	The *argument*s in any given *argument-list* need not all have the same type.

Return Type and Value: any, including array – The *argument* in *argument-list* that corresponds by position to *index*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if the value of *index* is not an index into *argument-list*, #VALUE! is returned.

[*Example*:

CHOOSE(E7,F7,G7,H7,I7,J7,K7,L7) results in Monday, when E7 contains 2, and the cells F7:L7 each contain the names of the week, from Sunday to Saturday

SUM(CHOOSE(E1,F20:G20,H20:J24)) results in the sum of the elements designated by F20:G20 or H20:J24, as determined by the value of E1

If B9:B11 contain 1, 3, and 3, respectively, and CHOOSE(B9:B11,10,20,30) is an array formula spanning 3 cells, the values of those 3 cells is 10, 30, and 30, respectively.

end example]

18.17.7.40 CLEAN

Syntax:

CLEAN (*string*)

Description:

Makes a string that is a copy of *string* with all so-called "non-printable" characters—those with internal values in the range U+0000–001F—removed.

Arguments:

Name	Type	Description
string	text	Designate the string to be cleaned.

Return Type and Value: text – The trimmed copy of *string*.

[*Example*:

CLEAN("A" & CHAR(2) & "BC") results in ABC, which is stored in A10
LEN(A10) results in 3

end example]

18.17.7.41 CODE

Syntax:

CODE (*string*)

Description: Determines the numeric code of the first character in *string*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
string	text	Designates a string containing one or more characters.

Return Type and Value: number – The numeric code of the first character in *string*.

However, if *string* is empty, #VALUE! is returned.

[*Example*:

`CODE("abc")` results in 97
`CODE(A10)` results in 97, when A1 contains `abc`

end example]

18.17.7.42 COLUMN

Syntax:

`COLUMN ( [` *reference* `] )`

Description: Finds the number of the column(s) corresponding to *reference*.

Arguments:

Name	Type	Description
reference	reference	A reference to a single cell or to a range of contiguous cells. If omitted, the behavior is as if *reference* referred to the cell containing the formula.

Return Type and Value: number – If *reference* refers to a single cell or to a single column of cells, the corresponding column is returned. If *reference* refers to a range of cells involving multiple columns, a horizontal array of the corresponding columns as numbers is returned.

However, if the range of cells referred to by *reference* is not contiguous, #REF! is returned.

[*Example*:

`COLUMN()` results in 4, when the cell containing the formula is in column 4
`COLUMN(E17:E19)` results in 5
`COLUMN(E16:F17)` results in a horizontal array containing 5and 6, respectively

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.43 COLUMNS

Syntax:

```
COLUMNS ( array )
```

Description: Finds the number of columns corresponding to *array*.

Arguments:

Name	Type	Description
array	array, reference	Any array.

Return Type and Value: number – The number of columns corresponding to *array*.

However, if the range of cells referred to by *array* is not contiguous, `#NULL!` is returned.

[*Example*:

`COLUMNS(E16:F16)` results in 2
`COLUMNS(E16:G18)` results in 3
`COLUMNS({1,2;3,4})` results in 2

end example]

18.17.7.44 COMBIN

Syntax:

```
COMBIN ( number , number-chosen )
```

Description: Computes the possible number of groups of size *number-chosen* that can be formed from *number* objects. [*Note*: A combination is any set or subset of objects, regardless of their internal order. Combinations are distinct from permutations, for which the internal order is significant. *end note*]

Mathematical Formula:

The number of combinations is as follows, where *number* = n and *number-chosen* = k:

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!\,(n-k)!}$$

where:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

where:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- *k* = argument *number-chosen*
- *n* = argument *number*

Arguments:

Name	Type	Description
number	number	The total number of objects available, truncated to an integer.
number-chosen	number	The number of objects in each combination, truncated to an integer.

Return Type and Value: number – The number of different combinations of *number-chosen* in *number*.

However, if

- *number* < 0, `#NUM!` is returned.
- *number-chosen* < 0, `#NUM!` is returned.
- *number* < *number-chosen*, `#NUM!` is returned.

[*Example*:

`COMBIN(8,2)` results in `28`
`COMBIN(10,4)` results in `210`
`COMBIN(6,5)` results in `6`

end example]

18.17.7.45 COMPLEX

Syntax:

`COMPLEX` (*real-number* , *imaginary-number* [, *suffix*])

Description: Makes a complex number in *x + yi* or *x + yj* text format from the arguments.

Arguments:

Name	Type	Description
real-number	number	The real number coefficient.
imaginary-number	number	The imaginary number coefficient.
suffix	text	"`i`" or "`j`". If omitted, "`i`" is used.

Return Type and Value: text – The complex number string specified by the arguments.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

If *real-number* has the value 0 and *imaginary-number* has a non-zero value, the resulting string contains just the real number. If *real-number* has a non-zero value and *imaginary-number* has a zero value, the resulting string contains just the imaginary number and suffix. If both *real-number* and *imaginary-number* have a zero value, the resulting string is `"0"`.

However, if *suffix* is neither `"i"` nor `"j"`, `#VALUE!` is returned.

[*Example*:

`COMPLEX(-3.5,19.6)` results in `-3.5+19.6i`
`COMPLEX(3.5,-19.6,"j")` results in `3.5-19.6j`
`COMPLEX(3.5,0)` results in `3.5`
`COMPLEX(0,2.4)` results in `2.4i`
`COMPLEX(0,0)` results in `0`

end example]

18.17.7.46 CONCATENATE

Syntax:

CONCATENATE (*argument-list*)

Description: Makes a string that is the concatenation of all the strings corresponding to the *arguments* in *argument-list*, taken left-to-right.

Arguments:

Name	Type	Description
argument-list	text	Each *argument* in *argument-list* shall designate a string.

Return Type and Value: text – The concatenated string.

[*Example*:

`CONCATENATE("text")` results in `text`
`CONCATENATE("The total is ",A10," units")` results in `The total is 43 units`, when `A10` contains `43`
`CONCATENATE(3," + ",4," = ",3+4)` results in `3 + 4 = 7`

end example]

18.17.7.47 CONFIDENCE

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

CONFIDENCE (*alpha* , *standard-dev* , *size*)

Description: Computes a value that can be used to construct a confidence interval for a population mean. The CONFIDENCE function assumes a normal distribution for calculation.

Arguments:

Name	Type	Description
alpha	number	The significance level used to compute the confidence level.
standard-dev	number	The population standard deviation for the data range.
size	number	The sample size, truncated to an integer.

Return Type and Value: number – A value that can be used to construct a confidence interval for a population mean.

However, if

- *alpha* ≤ 0 or *alpha* ≥ 1, #NUM! is returned.
- *standard-dev* ≤ 0, #NUM! is returned.
- *size* < 1, #NUM! is returned.

[*Example*:

CONFIDENCE(0.4,5,12) results in 1.214775614
CONFIDENCE(0.75,9,7) results in 1.083909234

end example]

18.17.7.48 CONVERT

Syntax:

CONVERT (*number* , *from-unit* , *to-unit*)

Description: Converts a number from one measurement system to another.

Arguments:

Name	Type	Description
number	number	The value to be converted from *from-unit*s to *to-unit*s.
from-unit	text	The unit to be converted from, where the permitted string values are shown in the tables below.
to-unit	text	The unit to be converted to, where the permitted string values are shown in the tables below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Weight and Mass	
Unit String	*Meaning*
g	Gram
lbm	Pound mass (avoirdupois)
ozm	Ounce mass (avoirdupois)
sg	Slug
u	U (atomic mass unit)

Distance	
Unit String	*Meaning*
ang	Angstrom
ft	Foot
in	Inch
m	Meter
mi	Statute mile
Nmi	Nautical mile
Pica	Point (1/72 inch)
yd	Yard

Time	
Unit String	*Meaning*
day	Day
hr	Hour
mn	Minute
sec	Second
yr	Year

Pressure	
Unit String	*Meaning*
at or atm	Atmosphere
mmHg	mm of Mercury
P or p	Pascal

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Force	
Unit String	*Meaning*
dy or dyn	Dyne
lbf	Pound force
N	Newton

Energy	
Unit String	*Meaning*
BTU or btu	BTU_{IT}
c	Thermodynamic calorie
cal	IT calorie
e	Erg
ev or eV	Electron volt
flb	Foot-pound
HPh or hh	Horsepower-hour
J	Joule
Wh or wh	Watt-hour

Power	
Unit String	*Meaning*
H or hp	Horsepower
W or w	Watt

Magnetism	
Unit String	*Meaning*
ga	Gauss
T	Tesla

Temperature	
Unit String	*Meaning*
C or cel	Degrees Celsius
F or fah	Degrees Fahrenheit
K or kel	Degrees Kelvin

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Liquid Measure			
Unit String	*Meaning*	*Family Conversion Factor (Informative)*	*Metric Conversion Factor (Informative)*
AU_tbs	Australian tablespoon		20 ml
cup	U.S. cup (reduced accuracy)	½ pt	236.59 ml +/-0.025%
CZ_mass	Czech mass		1.4147 l
CZ_mug	Czech mug	¼ CZ_mass	0.358 l
gal	U.S. gallon (reduced accuracy)	8 pt	3.7854 l +/-0.025% For the purpose of comparison with calculations by applications using the antecedents to ISO/IEC 29500, the specific metric conversion factor value of 3.78624 l might be appropriate.
GB_tbs or CA_tbs or JP_tbs	United Kingdom tablespoon		15 ml
imperial_gal or AU_gal or CA_gal or GB_gal	Imperial gallon	8 imperial_pt	4.54609 l
imperial_oz or AU_oz or CA_oz or GB_oz	Imperial fluid ounce	1/20 imperial_pt	28.4130625 ml
imperial_pt or AU_pt or CA_pt or GB_pt or uk_pt	Imperial pint		0.56826125 l uk_pt might be reduced accuracy: +/- 0.025%
imperial_qt or AU_qt or CA_qt or GB_qt	Imperial quart	2 imperial_pt	1.1365225 l
JP_cup	Japanese cup		200 ml
l or lt	Liter		
metric_cup or AU_cup or	Metric cup		250 ml

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Liquid Measure			
CA_cup or NZ_cup			
metric_tsp or AU_tsp or CA_tsp or JP_tsp or GB_tsp	Metric teaspoon		5 ml
ml	Milliliter	1/1000 l	1/1000 l
oz	U.S. fluid ounce (reduced accuracy)	1/16 pt	29.573 ml +/-0.025%
pt or us_pt	U.S. liquid pint		473.18 ml +/-0.025%
qt	U.S. liquid quart	2 pt	946.35 ml +/-0.025%
tbs	U.S. tablespoon (reduced accuracy)	½ oz	14.787 ml +/-0.025%
tsp	U.S. teaspoon (reduced accuracy)	1/6 oz	4.9289 ml +/-0.025%
US_cup	U.S. cup	½ US_pt	236.5882 ml
US_gal	U.S. gallon	8 US_pt	3.785412 l
US_oz	U.S. fluid ounce	1/16 US_pt	29.57353 ml
US_pt	U.S. liquid pint		473.1765 ml
US_qt	U.S. liquid quart	2 US_pt	946.3529 ml
US_tbs	U.S. tablespoon	½ US_oz	14.78676 ml
US_tsp	U.S. teaspoon	1/6 US_oz	4.928922 ml

The following abbreviated unit prefixes can be used with any metric unit:

Abbreviated Unit Prefixes	
Prefix String	*Meaning*
E	exa (1E+18)
P	peta (1E+15)
T	tera (1E+12)
G	giga (1E+09)
M	mega (1E+06)
k	kilo (1E+03)
h	hecto (1E+02)
e	deka (1E+01)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Abbreviated Unit Prefixes	
d	deci (1E-01)
c	centi (1E-02)
m	milli (1E-03)
u	micro (1E-06)
n	nano 1E-09)
p	pico (1E-12)
f	femto (1E-15)
a	atto (1E-18)

Unit names and prefixes are case-sensitive.

Return Type and Value: number – The value of *number* in *from-unit*s converted to *to-unit*s.

However, if

- The value of *from-unit* or *to-unit* is not one of the defined values, #N/A is returned.
- The *from-unit* and *to-unit* are from different measurement categories, #N/A is returned.
- The value of *from-unit* or *to-unit* has an abbreviated unit prefix, yet none is supported for that unit, #N/A is returned.

[*Example*:

```
CONVERT(10,"ozm","g") results in 283.4951521
CONVERT(1,"yd","mm") results in 914.4000003
CONVERT(1,"yd","cm") results in 91.44000003
CONVERT(1,"yd","m") results in 0.9144
CONVERT(1,"yd","km") results in 0.0009144
CONVERT(1,"mi","Nmi") results in 0.868976242
CONVERT(1,"day","sec") results in 86400
CONVERT(0,"K","C") results in -273.15
```

end example]

18.17.7.49 CORREL

Syntax:

CORREL (*array-1* , *array-2*)

Description: Computes the correlation coefficient of the two cell ranges designated by *array-1* and *array-2*.

Mathematical Formula:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The equation for the correlation coefficient is:

$$Correl(X,Y) = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(-\bar{x})^2 \sum(y-\bar{y})^2}}$$

where

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*array-1*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*array-2*)

Arguments:

Name	Type	Description
array-1	array, reference	The first cell range. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
array-2	array, reference	The second cell range. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The correlation coefficient of the cells in two cell ranges.

However, if

- *array-1* and *array-2* have a different number of data points, the return value is unspecified.
- *array-1* and *array-2* is empty, the return value is unspecified.
- The standard deviation of the values in *array-1* or *array-2* equals zero, the return value is unspecified.

[*Example*:

`CORREL({2.532,5.621;2.1,3.4},{5.32,2.765;5.2,6.7})` results in `-0.714976`

end example]

18.17.7.50 COS

Syntax:

COS (*x*)

Description: Computes the cosine of *x*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
x	number	The value, in radians, whose cosine is to be determined.

Return Type and Value: number – The cosine of *x*.

[*Example*:

`COS(-1)` results in `0.540302306`
`COS(0)` results in `1`
`COS(1)` results in `0.540302306`

end example]

18.17.7.51 COSH

Syntax:

COSH (*x*)

Description: Computes the hyperbolic cosine of *x*.

Arguments:

Name	Type	Description
x	number	The value whose hyperbolic cosine is to be determined.

Return Type and Value: number – The hyperbolic cosine of *x*.

However, if the magnitude of *x* is too large, `#NUM!` is returned.

[*Example*:

`COSH(-1)` results in `1.543080635`
`COSH(0)` results in `1`
`COSH(1)` results in `1.543080635`

end example]

18.17.7.52 COUNT

Syntax:

COUNT (*argument-list*)

Description: Counts the number of *arguments* in *argument-list* that contain numbers, and the number of cells referred to by *arguments* in *argument-list*, which contain numbers.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
argument-list	text	Each *argument* in *argument-list* designates a value. Arguments that are numbers, logical values, dates, or text representations of numbers shall be counted. If an argument is an array or reference, only numbers in that array or reference shall be counted. Empty cells, logical values, text, or error values in the array or reference shall be ignored. [*Note*: To count logical values, text, or error values as well, use the COUNTA (§18.17.7.53) function. *end note*]

Return Type and Value: number – The numeric argument and reference to numeric argument count.

[*Example*:

`COUNT(1,2,3,4,5)` results in 5
`COUNT({1,2,3,4,5})` results in 5
`COUNT({1,2,3,4,5},6,"7")` results in 7
`COUNT(10,E1)`, where E1 is an empty cell, results in 1, as E1 is ignored
`COUNT(10,E2)`, where E2 contains TRUE, results in 1, as E2 is ignored

end example]

18.17.7.53 COUNTA

Syntax:

COUNTA (*argument-list*)

Description: Counts the number of arguments that are not cell references, and the number of cells, referred to by arguments, which are not empty.

Arguments:

Name	Type	Description
argument-list	text	Each *argument* in *argument-list* designates a value. Arguments with values of any type shall be counted. However, empty cells shall not be counted. If an argument is an array or reference, only values in that array or reference shall be counted. Empty cells and text values in the array or reference shall be ignored. [*Note*: To exclude logical values, text, or error values, use the COUNT (§18.17.7.52) function. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The number of arguments that are not cell references, and the number of cells, referred to by arguments, which are not empty.

[*Example*:

`COUNTA(1,2,3,4,5)` results in 5
`COUNTA({1,2,3,4,5})` results in 15
`COUNTA({1,2,3,4,5},6,"7")` results in 7
`COUNTA(10,E1)`, where `E1` is an empty cell, results in 1, as `E1` is ignored
`COUNTA(10,E2)`, where `E2` contains TRUE, results in 2, as `E2` is counted

end example]

18.17.7.54 COUNTBLANK

Syntax:

`COUNTBLANK ( cell-range )`

Description: Counts the number of cells in a specified range of cells, which are empty. A cell containing a formula that returns an empty string is counted, whereas a cell containing a zero value is not.

Arguments:

Name	Type	Description
cell-range	reference	Designates the range of cells to be inspected.

Return Type and Value: number – The number of empty cells in the range specified.

[*Example*:

`COUNTBLANK(A2:C2)`, where `A2` and `B2` are empty, but `C2` is not, results in 2

end example]

18.17.7.55 COUNTIF

Syntax:

`COUNTIF ( cell-range , selection-criteria )`

Description: Counts the number of cells in a specified range of cells, whose values meet the specified criteria.

Arguments:

Name	Type	Description
cell-range	reference	Designates the range of cells to be inspected.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
selection-criteria	number, expression, reference, text	Designates the cells to be counted. In the case of text, *selection-criteria* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. *selection-criteria* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~).

Return Type and Value: number – The number of cells in the range specified that meet the criteria.

[*Example*: Given that A1, B1, C1, and D1, respectively, contain the values 3, 10, 7, and 10

`COUNTIF(A1:D1,"=10")` results in 2
`COUNTIF(A1:D1,">5")` results in 30
`COUNTIF(A1:D1,"<>10")` results in 2

Given that A2, B2, C2, and D2, respectively, contain the values `apples`, `oranges`, `grapes`, and `melons`

`COUNTIF(A2:D2,"*es")` results in 3
`COUNTIF(A2:D2,"??a*")` results in 2
`COUNTIF(A2:D2,"*l*")` results in 2

end example]

18.17.7.56 COUNTIFS

Syntax:

```
COUNTIFS ( count-range , cell-range-1 , selection-criteria-1
    [ , cell-range-2 , selection-criteria-2 [ , … ] ] )
```

Description: Counts the number of cells within a range that meet multiple criteria.

Arguments:

Name	Type	Description
count-range	reference	Designates the cells whose values are included. *count-range* does not have to have the same size and shape as *cell-range-1* through *cell-range-n*. The actual cells that are added are determined by using the top, left cell in *count-range* as the beginning cell, and then including cells that correspond in size and shape to *cell-range-1* through *cell-range-n*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
cell-range-1	reference	Designates the first range of cells to be inspected. Each cell in a range is counted only if all of the corresponding criteria specified are true for that cell.
selection-criteria-1	number, expression, reference, text	Designates the first range of cells to be counted. In the case of text, *selection-criteria-1* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. *selection-criteria* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~).
cell-range-n	reference	The optional arguments *selection-criteria-2* through *selection-criteria-n* have corresponding arguments *cell-range-2* through *cell-range-n*, and have the same semantics as *selection-criteria-1* and *cell-range-1*, respectively.
selection-criteria-n	number, expression, reference, text	

If a cell in any argument is an empty cell, it is treated as if it had the value 0.

Return Type and Value: number – The count of the cells corresponding to those selected.

[*Example*: Given the following data:

	A	B	C	D
1	Sales Person	Exceeded Tables Quota	Exceeded Chairs Quota	Exceeded Desks Quota
2	Davolio	Yes	No	No
3	Buchanan	Yes	Yes	No
4	Suyama	Yes	Yes	Yes
5	Leverling	No	Yes	Yes

`COUNTIFS(B2:D2,"=Yes")` results in 1 (counts how many times Davolio exceeded a sales quota for tables, chairs, and desks)

`COUNTIFS(B2:B5,"=Yes",C2:C5,"=Yes")` results in 2 (counts how many sales people exceeded both their tables and chairs quota)

`COUNTIFS(B5:D5,"=Yes",B3:D3,"=Yes")` results in 1 (counts how many times Leverling and Buchanan exceeded the same quota for tables, chairs, and desks)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.57 COUPDAYBS

Syntax:

COUPDAYBS (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the number of days from the beginning of the coupon period to the settlement date.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
frequency	number	The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		days as long as the first date was not 28 or 29 February, in which case it does not change.
		1 — Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2 — Actual/360. Similar to Basis 1, but only has 360 days per year.
		3 — Actual/365. Similar to Basis 1, but always has 365 days per year.
		4 — European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

Return Type and Value: number – The number of days from the beginning of the coupon period to the settlement date.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`COUPDAYBS(DATE(2007,1,25),DATE(2008,11,15),2,1)` results in `71`
`COUPDAYBS(DATE(2007,1,25),DATE(2008,11,15),2)` results in `70`

end example]

18.17.7.58 COUPDAYS

Syntax:

COUPDAYS (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the number of days in the coupon period that contains the settlement date.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
frequency	number	The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>frequency is truncated to an integer.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr>
<tr><td>2</td><td>Actual/360. Similar to</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The number of days in the coupon period that contains the settlement date.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.

 ©ISO/IEC 2016 – All rights reserved

- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`COUPDAYS(DATE(2007,1,25),DATE(2008,11,15),2,1)` results in `181`
`COUPDAYS(DATE(2007,1,25),DATE(2008,11,15),2)` results in `180`

end example]

18.17.7.59 COUPDAYSNC

Syntax:

COUPDAYSNC (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the number of days from the settlement date to the next coupon date.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
frequency	number	The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
`0` or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
			of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

Return Type and Value: number – The number of days from the settlement date to the next coupon date.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`COUPDAYSNC(DATE(2007,1,25),DATE(2008,11,15),2,1)` results in `110`
`COUPDAYSNC(DATE(2007,1,25),DATE(2008,11,15),2)` results in `110`

end example]

18.17.7.60 COUPNCD

COUPNCD (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the next coupon date after the settlement date.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
frequency	number	The number of coupon payments per year. For annual

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. frequency is truncated to an integer.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>days.</td></tr><tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The next coupon date after the settlement date, as a date.

However, if

©ISO/IEC 2016 – All rights reserved

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`COUPNCD(DATE(2007,1,25),DATE(2008,11,15),2,1)` results in `15-May-2007`

end example]

18.17.7.61 COUPNUM

COUPNUM (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the number of coupons payable between the settlement date and maturity date, rounded up to the nearest whole number.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
frequency	number	The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows: **Value** / **Day Count Basis** `0` or omitted / US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
			has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		• February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

Return Type and Value: number – The number of coupons payable between the settlement date and maturity date, rounded up to the nearest whole coupon.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`COUPNUM(DATE(2007,1,25),DATE(2008,11,15),2,1)` results in 4

end example]

18.17.7.62 COUPPCD

COUPPCD (*settlement* , *maturity* , *frequency* [, [*basis*]])

Description: Computes the previous coupon date before the settlement date.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>settlement</td><td>number</td><td>The security's settlement date.</td></tr>
<tr><td>maturity</td><td>number</td><td>The security's maturity date.</td></tr>
<tr><td>frequency</td><td>number</td><td>The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. frequency is truncated to an integer.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr><tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The previous coupon date before the settlement date, as a date.

©ISO/IEC 2016 – All rights reserved

However, if

- *settlement* or *maturity* is out of range for the current date system, #NUM! is returned.
- *settlement* ≥ *maturity*, #NUM! is returned.
- *frequency* is any number other than 1, 2, or 4, #NUM! is returned.
- *basis* < 0 or *basis* > 4, #NUM! is returned.

[*Example*:

COUPPCD(DATE(2007,1,25),DATE(2008,11,15),2,1) results in 15-Nov-2006

end example]

18.17.7.63 COVAR

Syntax:

COVAR (*array-1* , *array-2*)

Description: Computes covariance; that is, the average of the products of deviations for each data point pair in the two cell ranges designated by *array-1* and *array-2*.

Mathematical Formula:

The covariance is:

$$Cov(X,Y) = \frac{\sum(x-\bar{x})(y-\bar{y})}{n}$$

where

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*array-1*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*array-2*)

Arguments:

Name	Type	Description
array-1	number, name, array, reference to number	If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
array-2		

Return Type and Value: number – The covariance.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

However, if

- *array-1* and *array-2* have a different number of data points, the return value is unspecified.
- *array-1* or *array-2* is empty, the return value is unspecified.

[*Example*:

`COVAR({2.532,5.621;2.1,3.4},{5.32,2.765;5.2,6.7})` results in `-1.375374`

end example]

18.17.7.64 CRITBINOM

Syntax:

`CRITBINOM (` *number-trials* `,` *success-probability* `,` *alpha* `)`

Description: Computes the smallest value for which the cumulative binomial distribution is greater than or equal to a criterion value.

Arguments:

Name	Type	Description
number-trials	number	The number of Bernoulli trials.
success-probability	number	The probability of success on each trial.
alpha	number	The criterion value.

Return Type and Value: number – The smallest value for which the cumulative binomial distribution is greater than or equal to a criterion value.

However, if

- *number-trials* < 0, `#NUM!` is returned.
- *success-probability* is < 0 or *success-probability* > 1, `#NUM!` is returned.
- *alpha* < 0 or *alpha* > 1, `#NUM!` is returned.

[*Example*:

`CRITBINOM(6,0.5,0.75)` results in 4
`CRITBINOM(12,0.3,0.95)` results in 6

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.65 CUBEKPIMEMBER

Syntax:

```
CUBEKPIMEMBER ( connection , kpi-name , kpi-property [ , [ caption ] ] )
```

Description: Fetches from the OLAP cube on the OLAP server designated by *connection*, a Key Performance Indicator (KPI) name, property, and measure, and displays the name and property in the cell. A KPI is a quantifiable measurement, such as monthly gross profit or quarterly employee turnover, used to monitor an organization's performance.

Arguments:

Name	Type	Description
connection	text	The name of the connection to the cube.
kpi-name	text	The name of the KPI in the cube.
kpi-property	number	The KPI component to be returned, truncated to integer; it shall be one of the following:

Value	Description
1	The actual value, at the time the function is executed.
2	A target value of the KPI, which can be compared to the actual value in order to determine if the underlying indicator is meeting its goal.
3	The state of the KPI at a specific moment in time
4	A measure of the value over time
5	The relative importance assigned to the KPI on the server. If this KPI is assigned a parent KPI, then this number can be used on the server to proportionally adjust the results of this KPI value when calculating the value of the parent KPI. While this number can be shown in the spreadsheet application, it is assumed that it is used by the server for any calculations affecting the KPI.
6	The unique name of the member in the time dimension that defines the timeframe of the KPI.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		[*Example*: A KPI could be associated with the first quarter of the year 2007. *end example*] If 1 is specified, only *kpi-name* is displayed in the cell.
caption	text	An alternative string whose value is displayed in the cell instead of *kpi-name* and *kpi-property*.

Return Type and Value: any – The selected key performance indicator.

However, if

- *kpi-name* is not the name of a KPI in the cube, the return value is unspecified.
- *kpi-property* is outside the range 1–6, #N/A is returned.
- The *connection* name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified.

[*Example*:

```
CUBEKPIMEMBER("Sales","MySalesKPI",1)
CUBEKPIMEMBER("Sales","MySalesKPI",2,"Sales KPI Goal")
```

end example]

18.17.7.66 CUBEMEMBER

Syntax:

```
CUBEMEMBER ( connection , member-expression , [ , [ caption ] ] )
```

Description: Fetches from the OLAP cube on the OLAP server designated by *connection*, the member or tuple defined by *member-expression*. [*Note*: This function is used to ensure that the member or tuple exists in the cube. *end note*]

When a call to CUBEMEMBER is used as an argument to another CUBExxx function, the MDX expression that identifies the member or tuple is used by that CUBExxx function, rather than the displayed value in the cell of the CUBEMEMBER function.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
connection	text	The name of the connection to the cube.
member-expression	text, reference, array	A multidimensional expression (MDX) that evaluates to a unique member in the cube. Alternatively, *member-expression* can be a tuple, specified as a cell range or an array constant. [*Note*: MDX is a standard query language for OLAP cubes. *end note*]
caption	text	The string displayed in the cell instead of the caption from the cube (assuming it defines such a caption). When a tuple is returned, the caption used is the one for the last member in the tuple.

Return Type and Value: any – A member or tuple in a cube hierarchy.

However, if

- The connection name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified.
- At least one element within the tuple is invalid, the return value is unspecified.
- The syntax of *member-expression* is incorrect, the return value is unspecified.
- The member specified by *member-expression* doesn't exist in the cube, the return value is unspecified.
- The tuple is invalid because there is no intersection for the specified values, the return value is unspecified.
- The set contains at least one member with a different dimension than the other members, the return value is unspecified.

[*Example*:

```
CUBEMEMBER("Sales","[Time].[Fiscal].[2004]")
CUBEMEMBER($A$1,D$12)
CUBEMEMBER("Sales",(B4,C6,D5),"SalesFor2004")
CUBEMEMBER("Sales",{[Products].[Food];[Time].[Fiscal].[2004]})
CUBEMEMBER($A$1,C$12:D$12)
```

end example]

18.17.7.67 CUBEMEMBERPROPERTY

Syntax:

CUBEMEMBERPROPERTY (*connection* , *member-expression* , *property*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Fetches a property of a member in the OLAP cube on an OLAP server. [*Note*: Use this function o ensure that a member name exists within the cube and to return the specified property for this member. *end note*]

Arguments:

Name	Type	Description
connection	text	The name of the connection to the cube.
member-expression	text	A multidimensional expression (MDX) that evaluates to a unique member in the cube. [*Note*: MDX is a standard query language for OLAP cubes. *end note*]
property	text	The name of the property returned or a reference to a cell that contains the name of the property.

Return Type and Value: any – A property of a member in the OLAP cube.

However, if

- The connection name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified.
- The syntax of *member-expression* is incorrect, the return value is unspecified.
- The member specified by *member-expression* doesn't exist in the cube, the return value is unspecified.

[*Example*:

```
CUBEMEMBERPROPERTY("Sales","[Time].[Fiscal].[2004]",$A$3)
CUBEMEMBERPROPERTY("Sales","[Store].[MyFavoriteStore]",
  "[Store].[Store Name].[Store Sqft]")
```

end example]

18.17.7.68 CUBERANKEDMEMBER

Syntax:

```
CUBERANKEDMEMBER ( connection , set-expression , rank [ , caption ] )
```

Description: Fetches the n^{th}, or ranked, member in a set.

Arguments:

Name	Type	Description
connection	text	The name of the connection to the cube.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
set-expression	text	A set expression, such as "{[Item1].children}".
rank	number	Specifies the top value to return, truncated to integer. If 1, the top value is returned; if 2, the second-most top value is returned; and so on.
caption	text	The text displayed in the cell instead of the caption from the cube (assuming it defines such a caption).

Return Type and Value: any – The n[th] member in the set.

However, if

- The connection name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified. , the return value is unspecified.
- The syntax of *member-expression* is incorrect, the return value is unspecified.
- The set contains at least one member with a different dimension than the other members, the return value is unspecified.

[*Example*:

```
CUBERANKEDMEMBER("Sales",$D$4,1,"Top Month")
CUBERANKEDMEMBER("Sales",CUBESET("Sales","Summer","[2004].[June]",
  "[2004].[July]","[2004].[August]"),3,"Top Month")
```

end example]

18.17.7.69 CUBESET

Syntax:

```
CUBESET ( connection , set-expression [ , [ caption ][ , [ sort-order ]
   [ , [ sort-by ]]]] )
```

Description: Fetches from the OLAP cube on the OLAP server designated by *connection* the set of members or tuples that is defined by *set-expression*. [*Note*: Use this function to build dynamic reports that aggregate and filter data, by using the return value as a slicer in the CUBEVALUE function, the CUBERANKEDMEMBER function to choose specific members from the calculated set, and the CUBESETCOUNT function to control the size of the set. *end note*]

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>connection</td><td>text</td><td>The name of the connection to the cube.</td></tr>
<tr><td>set-expression</td><td>text, reference</td><td>A set expression that results in a set of members or tuples. set-expression can also be a cell reference to range that contains one or more members, tuples, or sets included in the set.</td></tr>
<tr><td>caption</td><td>text</td><td>The text displayed in the cell instead of the caption from the cube (assuming it defines such a caption).</td></tr>
<tr><td>sort-order</td><td>text</td><td>The type of sort, if any, to perform; it can be one of the following:
<table>
<tr><th>Value</th><th>Description</th><th>sort-by argument</th></tr>
<tr><td>0 or default</td><td>Leaves the set in existing order</td><td>Ignored</td></tr>
<tr><td>1</td><td>Sorts set in ascending order by sort_by</td><td>Required</td></tr>
<tr><td>2</td><td>Sorts set in descending order by sort_by</td><td>Required</td></tr>
<tr><td>3</td><td>Sorts set in alphabetic ascending order</td><td>Ignored</td></tr>
<tr><td>4</td><td>Sorts set in alphabetic descending order</td><td>Ignored</td></tr>
<tr><td>5</td><td>Sorts set in natural ascending order</td><td>Ignored</td></tr>
<tr><td>6</td><td>Sorts set in natural descending order</td><td>Ignored</td></tr>
</table>
An alphabetic sort for a set of tuples sorts on the last element in each tuple.</td></tr>
<tr><td>sort-by</td><td>text</td><td>The value by which to sort. [Example: To get the city with the highest sales, set-expression would be a set of cities, and sort-by would be the sales measure. To get the city with the highest population, set-expression would be a set of cities, and sort-by would be the population measure. end example]</td></tr>
</table>

Return Type and Value: any – The set of members or tuples.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if

- The connection name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified.
- The syntax of *member-expression* is incorrect, the return value is unspecified.
- The set contains at least one member with a different dimension than the other members, the return value is unspecified.
- *sort-order* is outside the range 0–6, #N/A is returned.
- *sort-order* requires *sort-by*, but *sort-by* is omitted, #VALUE! is returned.

[*Example*:

```
CUBESET("Finance","Order([Product].[Product].[Product Category]
.Members,[Measures].[Unit Sales],ASC)","Products")

CUBESET("Sales","[Product].[All Products].Children",
  "Products",1,"[Measures].[Sales Amount]")
```

end example]

18.17.7.70 CUBESETCOUNT

Syntax:

CUBESETCOUNT (*set*)

Description: Computes the number of items in a set.

Arguments:

Name	Type	Description
set	text	An expression that evaluates to a set defined by the CUBESET function.

Return Type and Value: number – The number of items in a set.

[*Example*:

```
CUBESETCOUNT(A3)
CUBESETCOUNT(CUBESET("Sales","[Product].[All Products].Children",
  "Products",1,"[Measures].[Sales Amount]"))
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.71 CUBEVALUE

Syntax:

CUBEVALUE (*connection* , *argument-list*)

Description: Fetches from the OLAP cube on the OLAP server designated by *connection*, the aggregated value defined by a series of member-expression *argument*s in *argument-list*.

Arguments:

Name	Type	Description
connection	text	The name of the connection to the cube.
argument-list	text, reference	Each *argument* in *argument-list* is text containing a multidimensional expression (MDX) that evaluates to a member or tuple within the cube. Alternatively, an *argument* can be a set defined with the CUBESET function. Use any *argument* as a slicer to define the portion of the cube for which the aggregated value is returned. If no measure is specified in an *argument*, the default measure for that cube is used. If a cell reference is used for an *argument*, and that cell reference contains a CUBE function, then that *argument* uses the MDX expression for the item in the referenced cell, and not the value displayed in that referenced cell. [*Note*: MDX is a standard query language for OLAP cubes. *end note*]

Return Type and Value: any – The aggregated value.

However, if

- The connection name is not a workbook connection stored in the workbook, the return value is unspecified.
- The OLAP server is not running, not available, or returns an error message, the return value is unspecified.
- At least one element within the tuple is invalid, the return value is unspecified.
- The syntax of *member-expression* is incorrect, the return value is unspecified.
- The member specified by an *argument* doesn't exist in the cube, the return value is unspecified.
- The tuple is invalid because there is no intersection for the specified values, the return value is unspecified. (This can occur with multiple elements from the same hierarchy.)
- The set contains at least one member with a different dimension than the other members, the return value is unspecified.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
CUBEVALUE("Sales","[Measures].[Profit]","[Time].[2004]",
  "[All Product].[Beverages]")
CUBEVALUE($A$1,"[Measures].[Profit]",D$12,$A23)
CUBEVALUE("Sales",$B$7,D$12,$A23)
```

end example]

18.17.7.72 CUMIPMT

Syntax:

CUMIPMT (*rate* , *nper* , *pv* , *start-period* , *end-period* , *type*)

Description: Computes the cumulative interest paid on a loan between *start-period* and *end-period*.

Arguments:

Name	Type	Description
rate	number	The interest rate.
nper	number	The total number of payment periods, truncated to integer.
pv	number	The present value.
start-period	number	The first period in the calculation. (Payment periods are numbered beginning with 1.)
end-period	number	The last period in the calculation.
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
0	Payment at the end of the period
1	Payment at the beginning of the period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The cumulative interest paid on a loan.

However, if

- *rate*, *nper*, or *pv* ≤ 0, #NUM! is returned.
- *start-period* < 1 or *end-period* < 1, or *start-period* > *end_period*, #NUM! is returned.
- *type* is any number other than 0 or 1, #NUM! is returned.

©ISO/IEC 2016 – All rights reserved

[*Example*:

`CUMIPMT(0.09/12,30*12,125000,13,24,0)` results in `-11135.23`
`CUMIPMT(0.09/12,30*12,125000,1,1,0)` results in `-937.50`

end example]

18.17.7.73 CUMPRINC

Syntax:

`CUMPRINC ( rate , nper , pv , start-period , end-period , type )`

Description: Computes the cumulative principal paid on a loan between *start-period* and *end-period*.

Arguments:

Name	Type	Description
rate	number	The interest rate.
nper	number	The total number of payment periods, truncated to integer.
pv	number	The present value.
start-period	number	The first period in the calculation. (Payment periods are numbered beginning with 1.)
end-period	number	The last period in the calculation.
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
`0`	Payment at the end of the period
`1`	Payment at the beginning of the period

Time information in the date arguments is ignored.

Return Type and Value: number – The cumulative principal paid on a loan.

However, if

- *rate*, *nper*, or *pv* ≤ 0, `#NUM!` is returned.
- *start-period* < 1 or *end-period* < 1, or *start-period* > *end_period*, `#NUM!` is returned.
- *type* is any number other than 0 or 1, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`CUMPRINC(0.09/12,30*12,125000,13,24,0)` results in `-934.11`
`CUMPRINC(0.09/12,30*12,125000,1,1,0)` results in `-68.28`

end example]

18.17.7.74 DATE

Syntax:

`DATE ( year , month , day )`

Description: Computes the serial date-time for the given date.

Arguments:

Name	Type	Description
year	number	A positive number, truncated to an integer representing the year, that together with *month* and *day* specifies the date whose serial date-time is to be computed. For the 1900 date system: • If *year* is in the range 0–99, inclusive, the year shall be interpreted as *year* + 1900. • If *year* is in the range 100–9999, inclusive, the year shall be interpreted as *year*. For the 1904 date system: • If *year* is in the range 0–1899, inclusive, the year shall be interpreted as *year* + 1900. • If *year* is in the range 1900–9999, inclusive, the year shall be interpreted as *year*.
month	number	A month, truncated to integer, that together with *year* and *day* specifies the date whose serial date-time is to be computed. *month* shall be interpreted as the number of months relative to the final month of the year prior to the specified year.
day	number	A day, truncated to integer, that together with *month* and *year* specifies the date whose serial date-time is to be computed. *day* shall be interpreted as the number of days relative to the last day of the month (and its associated year) prior to the month (and its associated year) as determined from *month* and *year* (see below).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The value of *month* or *day* in a *year-month-day* argument triplet can be out of range. *month* is simply an instance of counting a given number of months, minus one, relative to January of the year specified, using the Gregorian calendar [ISO 8601]. This calendar defines that there are 12 months in a year, and that when counting forward, the month following December of one year is January of the following year, and when counting backward, the month preceding January of one year is December of the previous year. Likewise, *day* is simply an instance of counting a given number of days, minus one, relative to the first day of the adjusted month, using the Gregorian calendar. This calendar defines the number of days in each month, and that when counting forward, the day following the final day of one month is the first day of the following month, and when counting backward, the day preceding the first day of one month is the final day of the previous month. [*Example*: The *year-month-day* argument triplets (2007, 12,32), (2007,13,1), and (2008,1,1) all result in the same serial date. *end example*]

[*Note*: One way to handle out-of-range values for *month* or *day* is as follows:

Compute yearAdjust = INT((*month* - 1)/12)
Compute adjustedMonth = *month* - (yearAdjust * 12)
Compute adjustedYear = *year* + yearAdjust.

A serialDateBase can now be computed for the first day of the adjustedYear and adjustedMonth. Finally, compute the serial date for the full triplet by adding (*day*-1) to this serialDateBase. *end note*]

Return Type and Value: number – The serial date-time for the given date.

However, if *year* is outside the acceptable range for the date system currently in use, `#NUM!` is returned.

[*Example*: For the 1900 backward compatibility date-base date system:

`DATE(0,1,1)` results in a serial date-time of `1`
`DATE(1899,1,1)` results in a serial date-time of `693598`
`DATE(1900,1,1)` results in a serial date-time of `1`
`DATE(9999,12,31)` results in a serial date-time of `2958465`

For the 1904 date system:

`DATE(4,1,1)` results in a serial date-time of `0`
`DATE(1899,1,1)` results in a serial date-time of `692136`
`DATE(1904,1,1)` results in a serial date-time of `0`
`DATE(9999,12,31)` results in a serial date-time of `2957003`

end example]

18.17.7.75 DATEDIF

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

DATEDIF (*start-date* , *end-date* , *unit*)

Description: Calculates the number of days, months, or years between two dates.

Arguments:

Name	Type	Description
start-date	number	The first date in the period, truncated to integer.
end-date	number	The last date in the period, truncated to integer.
unit	text	The count to be returned, as follows:

Value	Day Count Basis
"Y"	The number of complete years in the period.
"M"	The number of complete months in the period.
"D"	The number of days in the period.
"MD"	The difference between the days in *start-date* and *end-date*. The months and years of the dates are ignored.
"YM"	The difference between the months in *start-date* and *end-date*. The days and years of the dates are ignored.
"YD"	The difference between the days of *start-date* and *end-date*. The years of the dates are ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The number of days, months, or years between two dates, depending on the value of *unit*.

However, if

- *start-date* or *end-date* is out of range for the current date system, #NUM! is returned.
- *start-date* ≥ *end-date* #NUM! is returned.
- *unit* is any value other than those shown in the table above, #NUM! is returned.

[*Example*:

DATEDIF(DATE(2001,1,1),DATE(2003,1,1),"Y") results in 2 complete years

©ISO/IEC 2016 – All rights reserved

DATEDIF(DATE(2001,6,1),DATE(2002,8,15),"D") results in 440 days
DATEDIF(DATE(2001,6,1),DATE(2002,8,15),"YD") results in 75 days
DATEDIF(DATE(2001,6,1),DATE(2002,8,15),"MD") results in 14 days

end example]

18.17.7.76 DATEVALUE

Syntax:

DATEVALUE (*date-time-string*)

Description: Computes the serial date-time of the date represented by the string *date-time-string*, taking into account the current date system.

Arguments:

Name	Type	Description
date-time-string	text	The date and/or time whose serial date-time is to be computed. *date-time-string* can have any supported date and/or time format. If the year portion of *date-time-string* is omitted, the current year is used. Any time information in *date-time-string* shall be ignored. When *date-time-string* contains both a date and time part, times in *date-time-string* are truncated. Time-only values for *date-time-string* are special cased to return 0 so that when they are used in date addition and subtraction, time-only strings are ignored.

Return Type and Value: number – The serial date-time of the date represented by the string *date-time-string*.

However, if

- *date-time-string* is out of range for the current date system, #VALUE! is returned.
- *date-time-string* does not represent a date, #VALUE! is returned.
- *date-time-string* contains only a time, 0 is returned so that when it used in date addition and subtraction, time-only strings are ignored.

[*Example*: When the current year is 2006,

DATEVALUE("2/1/2006")
DATEVALUE("01-Feb-2006 10:06 AM")
DATEVALUE("2006/2/1")
DATEVALUE("2006-2-1")
DATEVALUE("1-Feb")

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

all result in 38749 for the 1900 date system, or 37287 for the 1904 date system. *end example*]

18.17.7.77 DAVERAGE

Syntax:

DAVERAGE (*database* , *field* , *criteria*)

Description: Averages the values in a column of a list or database that match the specified criteria.

In order to perform an operation on an entire column in a database, a blank line shall be entered below the column labels in the criteria range.

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database, which shall be a list of related data in which rows of related information are records, and columns of data are fields. The first row of the list shall contain labels for each column.
field	text, number	Indicates the column to which *criteria* shall be applied. It can either be a string containing the column's label, or the column's position number, where columns are numbered starting at 1. [*Example*: If column 3's label is "Age" then either 3 or "Age" can be used. *end example*]
criteria	reference	The range of cells that contains the specified conditions. Each cell in that range that contains a condition shall have a value that is the form of a number, an expression, a cell reference, or text that defines which cells are selected. In the case of text, *criteria* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~). A text *criteria* can also consist of any comparison operator followed by the operand against which each cell's value is to be compared. If the text does not begin with a comparison operator, the criteria matches any string starting with that text, as though the criteria were suffixed by an asterisk (*). [*Example*: A criteria of "Pea" can result in Pea, Pear, and Peach's being matched, whereas a criteria of "=Pea" only matches Pea. *end example*] Comparison operators do not require a prefix if used in the string, however it is not possible to specifically search for a string which begins with a comparison operator.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		The range shall include at least one column label and at least one cell below the column label in which a condition for the column is specified. [*Example*: If the range `G1:G2` contains the column label `Income` in `G1` and the amount 10,000 in `G2`, one could define the range as `MatchIncome` and use that name as *criteria*. *end example*] The value of *criteria* shall not overlap the range specified by *database*. To find rows that meet multiple criteria for a single column, all of the criteria shall be specified directly below one another in separate rows of the *criteria* range. To find rows that meet multiple criteria for multiple columns, all of the criteria shall be specified in the same row of the *criteria* range. To find rows that meet multiple criteria for multiple columns, where any criteria can be true, each of the criteria shall be specified in a different row of the *criteria* range. To find rows that meet multiple sets of criteria, where each set includes criteria for multiple columns, each set of criteria shall be specified in a separate row of the *criteria* range. To find rows that meet multiple sets of criteria, where each set includes criteria for one column, multiple columns with the same column heading shall be included in the *criteria* range.

Return Type and Value: number – The average of the values of the cells that correspond to the specified criteria.

[*Example*: Given the following data:

	A	B	C	D	E	F
1	Tree	Height	Age	Yield	Profit	Height
2	=Apple	>10				<16
3	=Pear					
4	Tree	Height	Age	Yield	Profit	
5	Apple	18	20	14	105.00	
6	Pear	12	12	10	96.00	
7	Cherry	13	14	9	105.00	
8	Apple	14	15	10	75.00	
9	Pear	9	8	8	76.80	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C	D	E	F
10	Apple	8	9	6	45.00	

the average yield of apple trees over 10 feet in height is computed by DAVERAGE(A4:E10,"Yield",A1:B2), which results in 12

The average age of all trees is computed by DAVERAGE(A4:E10,3,A4:E10), which results in 13

end example]

18.17.7.78 DAY

Syntax:

DAY (*date-value*)

Description: Computes the numeric day of the month in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*, taking into account the current date system.

Arguments:

Name	Type	Description
date-value	number, text	The date and/or time whose day is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its fractional part is ignored, or as a *string-constant* having any date and/or time format, in which case, any time information shall be ignored.

Return Type and Value: number – The day of the month in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*. The returned value shall be in the range 1–31.

However, if *date-value* is out of range for the current date system, #NUM! is returned.

[*Example*:

DAY(DATE(2006,1,2)) results in 2
DAY(DATE(2006,0,2)) results in 2
DAY(DATE(2013,9,0)) results in 31
DAY("2006/1/2 10:45 AM") results in 2
DAY(30000) results in 18 for the 1900 date system, or 19 for the 1904 date system

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.79 DAYS360

Syntax:

DAYS360 (*start-date* , *end-date* [, *method-flag*])

Description: Computes the signed number of days between two dates based on a 360-day year (twelve 30-day months).

Arguments:

Name	Type	Description
start-date	number	*start-date* and *end-date* are the dates for which the difference is to be computed. *start-date* can be earlier than, the same as, or later than *end-date*.
start-date	number	
method-flag	logical	Specifies whether to use the U.S. or European method in the calculation, as follows:

Value	Meaning
FALSE or omitted	U.S. (NASD) method: If the *start-date* is the 31st day of a month, it is changed to the 30th day of that same month. If the *end-date* is the 31st day of a month and the *start-date* is earlier than the 30th day of a month, the *end-date* is changed to the 1st day of the following month; otherwise the *end-date* is changed to the 30th day of the same month.
TRUE	European method: *start-date*s and *end-date*s that occur on the 31st day of a month are changed to the 30th day of the same month.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The signed number of days between two dates based on a 360-day year (12 30-day months). If *start-date* is later than *end-date*, the return value shall be negative, and the magnitude shall be the difference in days.

However, if *start-date* or *end-date* is out of range for the current date system, `#NUM!` is returned.

[*Example*:

`DAYS360(DATE(2002,2,3),DATE(2005,5,31))` results in `1198`
`DAYS360(DATE(2005,5,31),DATE(2002,2,3))` results in `-1197`
`DAYS360(DATE(2002,2,3),DATE(2005,5,31),FALSE)` results in `1198`

 ©ISO/IEC 2016 – All rights reserved

DAYS360(DATE(2002,2,3),DATE(2005,5,31),TRUE) results in 1197

⌋*end example*]

18.17.7.80 DB

Syntax:

DB (*cost* , *salvage* , *life* , *period* [, [*month*]])

Description: Computes the depreciation of an asset for a specified period using the fixed-declining balance method.

Mathematical Formula:

The fixed-declining balance method computes depreciation at a fixed rate. DB uses the following formulas to calculate depreciation for a period:

$$(cost - total\ depreciation\ from\ prior\ periods) \times rate$$

where:

$rate = 1 - \left(\frac{salvage}{cost}\right)^{\frac{1}{life}}$, rounded to three decimal places

Depreciation for the first and last periods is a special case. If argument *month* is omitted, depreciation for the first period is calculated using the formula cost × rate.

If month argument is entered, use the following formulas:

For the first period, DB uses this formula:

$$\frac{cost \times rate \times month}{12}$$

For the last period, DB uses this formula:

$$\frac{(cost - total\ depreciation\ from\ prior\ periods) \times rate \times (12 - month)}{12}$$

where:

- *cost* = argument *cost*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- *life* = argument *life*
- *rate* = 1 – (salvage / cost) ^ (1 / life), rounded to three decimals
- *salvage* = argument *salvage*
- *total depreciation from prior periods* = `DB(cost, salvage, life, 1, [month]) + DB(cost, salvage, life, 2, [month]) + … + DB(cost, salvage, life, period-1, [month]), where period > 1`

Arguments:

Name	Type	Description
cost	number	The initial cost of the asset.
salvage	number	The value at the end of the depreciation. (This is sometimes called the salvage value of the asset.)
life	number	The number of periods over which the asset is being depreciated. (This is sometimes called the useful life of the asset.)
period	number	The period for which the depreciation is to be calculated. (*period* shall use the same units as *life*.)
month	number	The number of months in the first year. If omitted, a value of 12 is used.

Return Type and Value: number – The depreciation of an asset for a specified period using the fixed-declining balance method.

However, if

- *cost*, *salvage*, *life*, or *period* < 0, `#NUM!` is returned.
- *month* is outside the range 1–12, `#NUM!` is returned.

[*Example*:

`DB(1000000,100000,6,1,7)` results in `186,083.33`
`DB(1000000,100000,6,2,7)` results in `259,639.42`
`DB(1000000,100000,6,3,7)` results in `176,814.44`
`DB(1000000,100000,6,4,7)` results in `120,410.64`
`DB(1000000,100000,6,5,7)` results in `81,999.64`
`DB(1000000,100000,6,6,7)` results in `55,841.76`
`DB(1000000,100000,6,7,7)` results in `15,845.10`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.81 DCOUNT

Syntax:

DCOUNT (*database* , *field* , *criteria*)

Description: Counts the number of values in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The count of the values of the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

For all the apple trees having a height between 10 and 16, the number of Age fields that contain numbers is computed by `DCOUNT(A4:E10,"Age",A1:F2)`, which results in `1`.

end example]

18.17.7.82 DCOUNTA

Syntax:

DCOUNTA (*database* , *field* , *criteria*)

Description: Counts the number of non-blank cells in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The count of the non-blank cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

For all the apple trees having a height between 10 and 16, the number of Profit fields that are not blank is computed by `DCOUNTA(A4:E10,"Profit",A1:F2)`, which results in `1`.

end example]

18.17.7.83 DDB

Syntax:

DDB (*cost* , *salvage* , *life* , *period* [, *factor*])

Description: Computes the depreciation of an asset for a specified period using the double-declining balance or some other specified method. [*Note*: Use VDB (§18.17.7.341) for a straight-line depreciation method when depreciation is greater than the declining balance calculation. *end note*]

Mathematical Formula:

$$MIN(\,(cost\ -total\ depreciation\ from\ prior\ periods) \times \left(\frac{factor}{life}\right), (cost\ -salvage\ -total\ depreiation\ from\ prior\ periods)\,)$$

where:

cost = argument *cost*

factor = argument *factor*

life = argument *life*

salvage = argument *salvage*

total depreciation from prior periods = `DDB(cost, salvage, life, 1, [factor]) + DDB(cost, salvage, life, 2, [factor]) + … + DDB(cost, salvage, life, period-1, [factor])`, where `period > 1`. Depreciation for period = 1 can be calculated by using cost × factor/life.

Arguments:

Name	Type	Description
cost	number	The initial cost of the asset.
salvage	number	The value at the end of the depreciation. (This is sometimes called the salvage value of the asset.)
life	number	The number of periods over which the asset is being

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		depreciated. (This is sometimes called the useful life of the asset.)
period	number	The period for which the depreciation is to be calculated. (*period* shall use the same units as *life*.)
factor	number	The rate at which the balance declines. If omitted, it is assumed to be 2 (the double-declining balance method).

Return Type and Value: number – The depreciation of an asset for a specified period.

However, if

- salvage < 0 #NUM! is returned.
- cost life ≤ 0, #NUM! is returned.
- life ≤ 0 #NUM! is returned.
- period ≤ 0, #NUM! is returned.
- factor ≤ 0, #NUM! is returned.

[*Example*:

`DDB(2400,300,10*365,1)` results in `1.32`
`DDB(2400,300,10*12,1,2)` results in `40.00`
`DDB(2400,300,10,1,2)` results in `480.00`
`DDB(2400,300,10,2,1.5)` results in `306.00`
`DDB(2400,300,10,10)` results in `22.12`

end example]

18.17.7.84 DEC2BIN

Syntax:

`DEC2BIN ( number [ , num-bin-digits ] )`

Description: Makes the binary equivalent of *number*, with the result having *num-bin-digits* digits.

Arguments:

Name	Type	Description
number	number	The decimal number that is to be converted to a binary string.
num-bin-digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-bin-digits* is ignored and the result has 10 digits. If

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		omitted, the minimum number of digits is used in the result. *num-bin-digits* is truncated to an integer.

Return Type and Value: text – The binary equivalent of *number* using twos-complement representation with the left-most bit (10th bit from the right) representing the sign bit.

However, if

- *number* is outside the range `-512` (1000000000 binary) to `511` (0111111111 binary), inclusive, `#NUM!` is returned.
- *number* needs more digits that *num-bin-digits*, `#NUM!` is returned.
- *num-bin-digits* ≤ 0 or > 10, `#NUM!` is returned.

[*Example*:

`DEC2BIN(23)` results in `10111`
`DEC2BIN(-256)` results in `1100000000`
`DEC2BIN(18,7)` results in `0010010`

end example]

18.17.7.85 DEC2HEX

Syntax:

`DEC2HEX ( number [ , num-hex-digits ] )`

Description: Makes the hexadecimal equivalent of *number*, with the result having *num-hex-digits* digits.

Arguments:

Name	Type	Description
number	number	The decimal number that is to be converted to a hexadecimal string.
num-bin-digits	number	*num-hex-digits* is the number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-hex-digits* is ignored and the result has 10 digits. If *num-hex-digits* is omitted, the minimum number of digits is used in the result. *num-hex-digits* is truncated to an integer.

Return Type and Value: text – The hexadecimal equivalent of *number* using twos-complement representation with the left-most bit (40th bit from the right) representing the sign bit.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if

- *number* is outside the range -549,755,813,888 (8000000000 hex) to 549,755,813,887 (7FFFFFFFFF hex), inclusive, #NUM! is returned.
- *number* needs more digits that *num-hex-digits*, #NUM! is returned.
- *num-hex-digits* ≤ 0 or > 10, #NUM! is returned.

[*Example*:

DEC2HEX(23) results in 17
DEC2HEX(-256) results in FFFFFFFF00
DEC2HEX(18,7) results in 0000012

end example]

18.17.7.86 DEC2OCT

Syntax:

DEC2OCT (*number* [, *num-oct-digits*])

Description: Makes the octal equivalent of *number*, with the result having *num-oct-digits* digits.

Arguments:

Name	Type	Description
number	number	The decimal number that is to be converted to an octal string.
num-bin-digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-oct-digits* is ignored and the result has 10 digits. If *num-oct-digits* is omitted, the minimum number of digits is used in the result. *num-oct-digits* is truncated to an integer.

Return Type and Value: text – The octal equivalent of *number* using twos-complement representation with the left-most bit (30th bit from the right) representing the sign bit.

However, if

- *number* is outside the range -536,870,912 (4000000000 octal) to 536,870,911 (3777777777 octal), inclusive, #NUM! is returned.
- *number* needs more digits that *num-oct-digits*, #NUM! is returned.
- *num-oct-digits* ≤ 0 or > 10, #NUM! is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

DEC2OCT(23) results in 27
DEC2OCT(-256) results in 7777777400
DEC2OCT(18,7) results in 0000022

end example]

18.17.7.87 DEGREES

Syntax:

DEGREES (*angle*)

Description: Converts *angle* in radians into degrees.

Arguments:

Name	Type	Description
angle	number	The number of radians that is to be converted into degrees.

Return Type and Value: number – *angle* in degrees.

[*Example*:

DEGREES(2 * PI()) results in 360
DEGREES(PI()) results in 180
DEGREES(PI()/2) results in 90
DEGREES(8.5) results in 487.0141259

end example]

18.17.7.88 DELTA

Syntax:

DELTA (*number-1* [, *number-2*])

Description: Compares two numbers for equality.

Arguments:

Name	Type	Description
number-1	number	The numbers that are to be compared for equality. If *number-2* is omitted, it is assumed to be zero.
number-2	number	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – 1 if *number-1* equals *number-2*; otherwise, 0.

[*Example*:

`DELTA(10.5,10.5)` results in `1`
`DELTA(10.5,10.6)` results in `0`
`DELTA(10.5)` results in `0`
`DELTA(0)` results in `1`

end example]

18.17.7.89 DEVSQ

Syntax:

DEVSQ (*argument-list*)

Description: Computes the sum of squares of deviations of data points from their sample mean.

Mathematical Formula:

$$DEVSQ = \sum (x - \bar{x})^2$$

where:

- x = each element in *argument-list*
- $\bar{x}$ = the mean of the elements in *argument-list*

Arguments:

Name	Type	Description
argument-list	logical, number, name, array, or reference to a number. Argument list can be a single argument that is an array or a reference to an array.	The *argument*s in *argument-list* designate the values for which the sum of squared deviations is to be calculated. Logical values and text representations of numbers occurring directly in the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The sum of squares of deviations of data points from their sample mean.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

DEVSQ(5.6,8.2,9.2) results in 6.906666667
DEVSQ({5.6,8.2,9.2}) results in 6.906666667

end example]

18.17.7.90 DGET

Syntax:

DGET (*database* , *field* , *criteria*)

Description: Extracts a single value from a column of a list that matches the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The value of the cell that corresponds to the specified criteria.

However, if

- No record matches the criteria, #VALUE! is returned.
- More than one record matches the criteria, #NUM! is returned.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

For all the apple trees having a height between 10 and 16, the number of Profit fields that are not blank is computed by DGET(A4:E7,"Yield",A1:A2), which results in 14.

end example]

18.17.7.91 DISC

Syntax:

DISC (*settlement* , *maturity* , *pr* , *redemption* [, [*basis*]])

Description: Computes the discount rate for a security.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Mathematical Formula:

$$DISC = \frac{redemption - par}{par} \times \frac{B}{SM}$$

where:

- B = number of days in a year, depending on the year basis.
- DSM = number of days between settlement and maturity.
- par = argument *pr*
- $redemption$ = argument *redemption*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
pr	number	The security's price per 100 currency units face value.
redemption	number	The security's redemption value per 100 currency units face value.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
`0` or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1 — Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2 — Actual/360. Similar to Basis 1, but only has 360 days per year.
		3 — Actual/365. Similar to Basis 1, but always has 365 days per year.
		4 — European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		changed to day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

The currency units of *pr* and *redemption* are assumed to be the same currency.

Return Type and Value: number – The discount rate for a security.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *pr* or *redemption* ≤ 0, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`DISC(DATE(2007,1,25),DATE(2007,6,15),97.975,100,1)` results in `5.2420%`

end example]

18.17.7.92 DMAX

Syntax:

DMAX (*database* , *field* , *criteria*)

Description: Computes the maximum value of the cells in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The maximum of the values of the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The maximum profit of apple and pear trees is computed by `DMAX(A4:E10,"Profit",A1:A3)`, which results in `105`.

end example]

18.17.7.93 DMIN

Syntax:

`DMIN ( database , field , criteria )`

Description: Computes the minimum value of the cells in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The minimum of the values of the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The minimum profit of apple trees over 10 in height is computed by `DMIN(A4:E10,"Profit",A1:B2)`, which results in `75`.

end example]

18.17.7.94 DOLLAR

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

DOLLAR (*number* [, *num-decimal*])

Description: Produces a string containing *number* rounded to *num-decimal* decimal places. The formatting applied to the string for the thousands separator, radix point, and currency symbol are implementation-specific.

Arguments:

Name	Type	Description
number	number	The number that is to be formatted.
num-decimal	number	Designate the number of decimal places to be used in the resulting string; it is truncated to an integer. If *num-decimal* is negative, *number* is rounded to the left of the decimal point. If omitted, a value of 2 shall be assumed.

Return Type and Value: text – The string containing *number* rounded to *num-decimal* decimal places, and have a currency symbol and thousands separators.

[*Example*: When the spreadsheet application defines the currency symbol to be "$", the thousands separator to be "," and the radix to be ".":

DOLLAR(1234.567) results in $1,234.57
DOLLAR(1234.567,-2) results in $1,200
DOLLAR(-1234.567,4) results in ($1,234.5670)

When the spreadsheet application defines the currency symbol to be "€", the thousands separator to be " " and the radix to be ",":

DOLLAR(1234.567) results in 1 234,57 €
DOLLAR(1234.567,-2) results in 1 200 €
DOLLAR(-1234.567,4) results in -1 234,5670 €

When the spreadsheet application defines the currency symbol to be "SFr.", the thousands separator to be "'" and the radix to be ".":

DOLLAR(1234.567) results in SFr. 1'234.57
DOLLAR(1234.567,-2) results in SFr. 1'200
DOLLAR(-1234.567,4) results in SFr. -1'234.5670

When the spreadsheet application defines the currency symbol to be "kr", the thousands separator to be " " and the radix to be ",":

DOLLAR(1234.567) results in kr 1 234,57
DOLLAR(1234.567,-2) results in kr 1 200

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

DOLLAR(-1234.567,4) results in kr -1 234,5670
end example]

18.17.7.95 DOLLARDE

Syntax:

DOLLARDE (*fractional-dollar* , *fraction*)

Description: Converts a fractional dollar price into a dollar price expressed as a decimal number. [*Note*: Fractional dollar numbers are sometimes used for securities prices. *end note*] The fractional part of *fractional-dollar* is scaled to match the magnitude of *fraction* by moving the decimal place right by the number of digits in *fraction*. The *fractional-dollar* m.n is computed into a decimal dollar value as m + i, where i is an intermediate result equal to (((0.n)*(10x))/fraction), and x is the base 10 log of fraction, rounded up to the nearest whole number.

[*Example*: Given a *fractional-dollar* value of 1.02 and a *fraction* value of 16, the fractional part of *fractional-dollar* is multiplied by 100, giving a value of 2 to be divided by *fraction* before being added to the integral part of *fractional-dollar*, yielding a decimal price of 1.125. *end example*]

Arguments:

Name	Type	Description
fractional-dollar	number	The number to be interpreted as a fractional dollar price.
fraction	number	The integer to use in the denominator of the fraction.

Return Type and Value: number – The dollar price expressed as a decimal number.

However, if

- *fraction* < 0, #NUM! is returned.
- *fraction* = 0, #DIV/0! is returned.

[*Example*:

DOLLARDE(1.02,16) results in 1.125
DOLLARDE(1.1,32) results in 1.3125

end example]

18.17.7.96 DOLLARFR

Syntax:

DOLLARFR (*decimal-dollar* , *fraction*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Converts a dollar price expressed as a decimal into a dollar price expressed as a fraction. This function is used to convert decimal dollar numbers, such as securities prices, to fractional numbers.

Arguments:

Name	Type	Description
decimal-dollar	number	The number expressed as a decimal.
fraction	number	The integer to use in the denominator of the fraction.

Return Type and Value: number – The dollar price expressed as a fractional number. [*Example*: A result of *m.n* means *m* + *n*/*fraction* dollars. *end example*] The fractional part of the return value is scaled to have the same number of digits after the decimal point, as there are digits in *fraction*. [*Example*: DOLLARFR(1.125, 16) has a two-digit *fraction* value and so returns the two-digit fractional number 1.02 *end example*] If an exact numerator cannot be found, the function returns the lowest numerator that could be used with *fraction*, multiplied by a power of ten. [*Example*: DOLLARFR(1.5,3) returns 1.15, as there is no exact fraction which satisfies n/3 = 0.5, and 15/30 represents the lowest power of ten *fraction* can be multiplied by to obtain an exact value. *end example*]

However, if

- *fraction* < 0, `#NUM!` is returned.
- *fraction* = 0, `#DIV/0!` is returned.

[*Example*:

`DOLLARFR(1.125,16)` results in `1.02`
`DOLLARFR(1.125,32)` results in `1.04`

end example]

18.17.7.97 DPRODUCT

Syntax:

DPRODUCT (*database* , *field* , *criteria*)

Description: Computes the product of the values of the cells in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The product of the values of the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The product of the yields from apple trees with a height greater than 10 is computed by `DPRODUCT(A4:E10,"Yield",A1:B2)`, which results in `140`.

end example]

18.17.7.98 DSTDEV

Syntax:

DSTDEV (*database* , *field* , *criteria*)

Description: Estimates the standard deviation of a population based on a sample by using the numbers in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – An estimate of the standard deviation of a population based on the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The estimated standard deviation in the yield of apple and pear trees if the data in the database is only a sample of the total orchard population is computed by `DSTDEV(A4:E10,"Yield",A1:A3)`, which results in `2.97`.

end example]

18.17.7.99 DSTDEVP

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

DSTDEVP (*database* , *field* , *criteria*)

Description: Computes the standard deviation of a population based on the entire population by using the numbers in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The standard deviation of a population based on the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The true standard deviation in the yield of apple and pear trees if the data in the database is the entire population is computed by `DSTDEVP(A4:E10,"Yield",A1:A3)`, which results in `2.65`.

end example]

18.17.7.100 DSUM

Syntax:

DSUM (*database* , *field* , *criteria*)

Description: Computes the sum of the values in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The sum of the values of the cells that correspond to the specified criteria.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The total profit from apple trees is computed by `DSUM(A4:E10,"Profit",A1:A2)`, which results in 225.

The total profit from apple trees with a height between 10 and 16 is computed by `DSUM(A4:E10,"Profit",A1:F2)`, which results in 75.

end example]

18.17.7.101 DURATION

Syntax:

DURATION (*settlement* , *maturity* , *coupon* , *yld* , *frequency* [, [*basis*]])

Description: Computes the Macaulay duration for an assumed par value of 100. Duration is defined as the weighted average term to maturity of the cash flows from a bond. The weight of each cash flow is determined by dividing the present value of the cash flow by the price.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
coupon	number	The security's annual coupon rate.
yld	number	The security's annual yield.
frequency	number	The number of coupon payments per year. For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4. *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
`0` or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description		
				• For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
			1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
			2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			3	Actual/365. Similar to Basis 1, but always has 365 days per year.
			4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

Return Type and Value: number – The Macaulay duration for an assumed par value of 100.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *coupon* or *yld* < 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`DURATION(DATE(2008,1,1),DATE(2016,1,1),0.08,0.09,2,1)` results in `5.993774956`

end example]

18.17.7.102 DVAR

Syntax:

DVAR (*database* , *field* , *criteria*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Estimates the variance of a population based on a sample by using the numbers in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – An estimate of the variance of a population based on the cells that correspond to the specified criteria.

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The estimated variance in the yield of apple and pear trees if the data in the database is only a sample of the total orchard population is computed by `DVAR(A4:E10,"Yield",A1:A3)`, which results in `8.8`.

end example]

18.17.7.103 DVARP

Syntax:

DVARP (*database* , *field* , *criteria*)

Description: Calculates the variance of a population based on the entire population by using the numbers in a column of a list or database that match the specified criteria. (See the DAVERAGE function §18.17.7.77.)

Arguments:

Name	Type	Description
database	reference	The range of cells that makes up the list or database.
field	text, number	The column to which *criteria* shall be applied.
criteria	reference	The range of cells that contains the specified conditions.

For a detailed description of each argument, see the DAVERAGE function §18.17.7.77.

Return Type and Value: number – The variance of a population based on the entire population using the cells that correspond to the specified criteria.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Using the data in the example in the DAVERAGE function §18.17.7.77:

The true variance in the yield of apple and pear trees if the data in the database is the entire orchard population is computed by DVARP(A4:E10,"Yield",A1:A3), which results in 7.04.

end example]

18.17.7.104 ECMA.CEILING

Syntax:

ECMA.CEILING (*x* , *significance*)

Description: Computes a value that is *x* rounded-up, away from zero, to the nearest multiple of *significance*. Regardless of the sign of *x*, a value is rounded up when adjusted away from zero.

Arguments:

Name	Type	Description
x	number	The value to be rounded.
significance	number	The multiple to which *x* is to be rounded. If *x* is negative, and *significance* is negative, then the value is rounded down (away from zero). If *x* is negative, and *significance* is positive, then the value is rounded up, towards zero.

Return Type and Value: number – The rounded-up value of *x*.

However, if

- *x* is positive and *significance* is negative, #NUM! is returned.
- *x* and/or *significance* is zero, zero is returned.

[*Example*:

ECMA.CEILING(4.3,2) rounds 4.3 up to nearest multiple of 2; that is, to 6
ECMA.CEILING(4.3,-2) rounds 4.3 up to nearest multiple of -2; that is, to #NUM!
ECMA.CEILING(-4.3,2) rounds -4.3 up to the nearest multiple of 2; that is, to -4
ECMA.CEILING(-4.3,-2) rounds -4.3 up to the nearest multiple of -2; that is, to -6

end example]

18.17.7.105 EDATE

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

EDATE (*start-date* , *month-offset*)

Description: Computes the serial date-time of the date that is *month-offset* months from the date specified by the date *date-string*, taking into account the current date system.

Arguments:

Name	Type	Description
start-date	number	The start date.
month-offset	number	The number of months before or after *start-date*, truncated to integer. A positive value yields a future date; a negative value yields a past date; a zero value yields the date *start-date*.

Return Type and Value: number – The serial date-time of the date that is *month-offset* months from the date specified by the date *date-string*, as a whole number.

However, if

- *start-value* is out of range for the current date system, `#NUM!` is returned.
- *start-value* plus *month-offset* is out of range for the current date system, `#NUM!` is returned.

[*Example*: For the 1900 date system:

`EDATE(DATE(2006,1,31),5)` results in a serial date-time of `38898`
`EDATE(DATE(2004,2,29),12)` results in a serial date-time of `38411`
`EDATE(DATE(2004,2,28),12)` results in a serial date-time of `38411`
`EDATE(DATE(2004,1,15),-23)` results in a serial date-time of `37302`

For the 1904 date system:

`EDATE(DATE(2006,1,31),5)` results in a serial date-time of `37436`
`EDATE(DATE(2004,2,29),12)` results in a serial date-time of `36949`
`EDATE(DATE(2004,2,28),12)` results in a serial date-time of `36949`
`EDATE(DATE(2004,1,15),-23)` results in a serial date-time of `35840`

end example]

18.17.7.106 EFFECT

Syntax:

EFFECT (*nominal-rate* , *npery*)

Description: Computes the effective annual interest rate, given the nominal annual interest rate and the number of compounding periods per year.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Mathematical Formula:

$$EFFECT = \left(1 + \frac{Nominal_rate}{Npery}\right)^{Npery} - 1$$

where:

- *Nominal_rate* = argument *nominal-rate*
- *Npery* = argument *npery*

Arguments:

Name	Type	Description
nominal-rate	number	The nominal interest rate.
npery	number	The number of compounding periods per year, truncated to integer.

Return Type and Value: number – The effective annual interest rate.

However, if

- *nominal-rate* ≤ 0, `#NUM!` is returned.
- *npery* < 1, `#NUM!` is returned.

[*Example*:

`EFFECT(0.0525,4)` results in `5.3543%`

end example]

18.17.7.107 EOMONTH

Syntax:

`EOMONTH` (*start-date* , *month-offset*)

Description: Computes the serial date-time of the last day of the month for the date that is *month-offset* months from the date specified by the date *start-date*, taking into account the current date system.

Arguments:

Name	Type	Description
start-date	number	The start date.
month-offset	number	The number of months before or after *start-date*, truncated to integer. A positive value yields a future date; a negative value yields a past date; a zero value yields the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		date *start-date*.

Return Type and Value: number – The serial date-time of the last day of the month for the date that is *month-offset* months from the date specified by the date *start-date*, as a whole number.

However, if

- *start-date* is not a date, `#NUM!` is returned.
- *start-date* plus *month-offset* does not yield a date, `#NUM!` is returned.

[*Example*: For the 1900 date system:

`EOMONTH(DATE(2006,1,31),5)` results in a serial date-time of 38898
`EOMONTH(DATE(2004,2,29),12)` results in a serial date-time of 38411
`EOMONTH(DATE(2004,2,28),12)` results in a serial date-time of 38411
`EOMONTH(DATE(2004,1,15),-23)` results in a serial date-time of 37315

For the 1904 date system:

`EOMONTH(DATE(2006,1,31),5)` results in a serial date-time of 37436
`EOMONTH(DATE(2004,2,29),12)` results in a serial date-time of 36949
`EOMONTH(DATE(2004,2,28),12)` results in a serial date-time of 36949
`EOMONTH(DATE(2004,1,15),-23)` results in a serial date-time of 35853

end example]

18.17.7.108 ERF

Syntax:

`ERF ( lower-bound [ , upper-bound ] )`

Description: Computes the error function integrated between *lower-bound* and *upper-bound*.

Mathematical Formula:

If *upper-bound* is omitted:

$$ERF(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt$$

where:

- z = argument *lower-bound*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If *upper-bound* is present:

$$ERF(a,b) = \frac{2}{\sqrt{\pi}} \int_{a}^{b} e^{-t^2}\, dt = ERF(b) - ERF(a)$$

where:

- b = argument *upper-bound*
- a = argument *lower-bound*

Arguments:

Name	Type	Description
lower-bound	number	The lower bound for integrating ERF.
upper-bound	number	The upper bound for integrating ERF. If omitted, the value of the upper bound is *lower-bound*, and the lower bound becomes zero.

Return Type and Value: number – The error function integrated between *lower-bound* and *upper-bound*.

However, if

- *lower-bound* is negative, `#NUM!` is returned.
- *upper-bound* is negative, `#NUM!` is returned.

[*Example*:

`ERF(1.234,4.5432)` results in `0.08096060`
`ERF(0,1.345)` results in `0.94284416`
`ERF(0,1.345)` results in `0.94284416`

end example]

18.17.7.109 ERFC

Syntax:

ERFC (*lower-bound*)

Description: Computes the complementary error function integrated between *lower-bound* and ∞.

Mathematical Formula:

$$ERFC(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2}\, dt = 1 - ERF(x)$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

where:

- *x* = argument *lower-bound*

Arguments:

Name	Type	Description
lower-bound	number	The lower bound for integrating ERFC.

Return Type and Value: number – The complementary error function integrated between *lower-bound* and ∞.

However, if *lower-bound* is negative

lower-bound or *upper-bound* is negative, `#NUM!` is returned.

[*Example*:

`ERFC(1.234)` results in `0.08096060`
`ERFC(0)` results in `1.00000000`

end example]

18.17.7.110 ERROR.TYPE

Syntax:

`ERROR.TYPE ( value )`

Description: Determines the kind of the error value designated by *value*.

Arguments:

Name	Type	Description
value	any	A value whose type is to be determined. No conversion shall take place on the argument passed to this function.

Return Type and Value: number – The kind of the error value designated by *value*, as follows:

value	Return Value
`#NULL!`	1
`#DIV/0!`	2
`#VALUE!`	3
`#REF!`	4

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

value	Return Value
#NAME?	5
#NUM!	6
#N/A	7
#GETTING_DATA	8
Anything else	#N/A

[*Example*:

ERROR.TYPE(A1) results in 2 if A1 evaluates to #DIV/0!
ERROR.TYPE(A1) results in 4 if A1 evaluates to #REF/0!
ERROR.TYPE(A1) results in 7 if A1 evaluates to #N/A
ERROR.TYPE(A1) results in #N/A if A1 evaluates to a non-error value, such as a number or text

end example]

18.17.7.111 EVEN

Syntax:

EVEN (*x*)

Description: Computes *x* rounded to the nearest even integer, away from zero. Regardless of the sign of *x*, a value is rounded up when adjusted away from zero.

Arguments:

Name	Type	Description
x	number	The value to be rounded.

Return Type and Value: number – The rounded value of *x*. If *x* is zero, the result is zero.

[*Example*:

EVEN(1.5) rounds 1.5 up to the nearest even integer; that is, to 2.
EVEN(3) rounds 3 up to the nearest even integer; that is, to 4.
EVEN(2) rounds 2 up to the nearest even integer; that is, to 2.
EVEN(-1) rounds -1 up to the nearest even integer; that is, to -2.

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.112 EXACT

Syntax:

EXACT (*string-1* , *string-2*)

Description: Performs a case-sensitive, character-by-character, lexical comparison of *string-1* and *string-2*.

Arguments:

Name	Type	Description
string-1	text	The two strings to be compared.
string-2	text	

Return Type and Value: logical – TRUE if *string-1* and *string-2* have the exact same length and contents; otherwise, FALSE.

[*Example*:

EXACT("ABC","ABC") results in TRUE
EXACT("ABC","ABCD") results in FALSE
EXACT("Abc","aBC") results in FALSE
EXACT("","") results in TRUE

end example]

18.17.7.113 EXP

Syntax:

EXP (*x*)

Description: Computes e^x, where the constant *e* is the base of the natural logarithm.

Arguments:

Name	Type	Description
x	number	The exponent to which *e* is to be raised.

Return Type and Value: number – e^x.

However, if *x* is too large for the result to be representable, #NUM! is returned.

[*Example*:

EXP(-1) results in 0.367879441

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

EXP(0) results in 1
EXP(1) results in 2.718281828
EXP(2) results in 7.389056099

end example]

18.17.7.114 EXPONDIST

Syntax:

EXPONDIST (*x* , *lambda* , *cumulative-flag*)

Description: Computes the exponential distribution.

Mathematical Formula:

The equation for the probability density function is:

$f(x,\lambda) = \lambda e^{-\lambda x}$

The equation for the cumulative distribution function is:

$F(x,\lambda) = 1 - e^{-\lambda x}$

where:

- *x* = argument *x*
- λ = argument *lambda*

Arguments:

Name	Type	Description
x	number	The value at which the function is evaluated.
lambda	number	The inverse of the mean.
cumulative-flag	logical	Determines the form of the function. If TRUE, EXPONDIST returns the cumulative distribution function; if FALSE, EXPONDIST returns the probability density function.

Return Type and Value: number – The exponential distribution.

However, if

- *x* < 0, #NUM! is returned.
- *lambda* ≤ 0, #NUM! is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`EXPONDIST(0.2,10,FALSE)` results in `1.353352832`
`EXPONDIST(2.3,1.5,TRUE)` results in `0.968254364`

end example]

18.17.7.115 FACT

Syntax:

FACT (*x*)

Description: Computes the factorial of *x*.

Arguments:

Name	Type	Description
x	number	The non-negative value whose factorial is to be computed. *x* is truncated to an integer.

Return Type and Value: number – The factorial of *x*.

However, if

- *x* is negative, `#NUM!` is returned.
- *x* is too large for the result to be representable, `#NUM!` is returned.

[*Example*:

`FACT(5)` results in `120`
`FACT(3.5)` results in `6`
`FACT(0)` results in `1`

end example]

18.17.7.116 FACTDOUBLE

Syntax:

FACTDOUBLE (*n*)

Description: Computes the double factorial of *n*.

Mathematical Formula:

If *n* is even:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

$n!! = n(n-2)(n-4)\ldots(4)(2)$

If *n* is odd:

$n!! = n(n-2)(n-4)\ldots(3)(1)$

where:

- *n* = argument *n*

Arguments:

Name	Type	Description
n	number	The non-negative value whose double factorial is to be computed. *n* is truncated to an integer.

Return Type and Value: number – The double factorial of *n*.

However, if

- *n* is negative, `#NUM!` is returned.
- *n* is too large for the result to be representable, `#NUM!` is returned.

[*Example*:

`FACTDOUBLE(5)` results in `15`
`FACTDOUBLE (3.5)` results in `3`
`FACTDOUBLE (0)` results in `1`

end example]

18.17.7.117 FALSE

Syntax:

```
FALSE ( )
```

Description: Computes the value FALSE. (A call to function FALSE is equivalent to using the *logical-constant* FALSE.)

Arguments: None.

Return Type and Value: logical – The value FALSE.

[*Example*:

`FALSE()` results in FALSE

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.118 FDIST

Syntax:

`FDIST` (*x* , *degrees-freedom-1* , *degrees-freedom-2*)

Description: Computes the F probability distribution.

Mathematical Formula:

$FDIST = P(F > x)$

where:

- F is a random variable that has an F distribution with degrees of freedom *degrees-freedom-1* and *degrees-freedom-2*
- x is argument *x*

Arguments:

Name	Type	Description
x	number	The value at which the function is to be evaluated.
degrees-freedom-1	number	The number of degrees of freedom for the numerator, truncated to an integer.
degrees-freedom-2	number	The number of degrees of freedom for the denominator, truncated to an integer.

Return Type and Value: number – The F probability distribution.

However, if

- *x* is negative, `#NUM!` is returned.
- *degrees-freedom-1* < 1 or *degrees-freedom-1* ≥ 10^{10}, `#NUM!` is returned.
- *degrees-freedom-2* < 1 or *degrees-freedom-2* ≥ 10^{10}, `#NUM!` is returned.

[*Example*:

`FDIST(12.345,3,4)` results in `0.017226183`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.119 FIND

Syntax:

```
FIND ( string-1 , string-2 [ , start-pos ] )
```

Description: Performs a case-sensitive search using a lexical comparison for the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. (FIND is intended for use with languages that use the single-byte character set (SBCS), whereas FINDB (§18.17.7.120) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string-1	text	Designate the string to be searched for within the string designated by *string-2*.
string-2	text	
start-pos	number	The number of the start position within *string-2* for which *string-1* is to be searched. The start position of the first character is 1. If omitted, a position of 1 shall be assumed. *start-pos* shall be at least 1.

Return Type and Value: number – The start position of the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. If *string-1* is an empty string, it shall always be found in any *string-2* at position *start-pos*, or at position 1 if *start-pos* is omitted.

However, if

- *string-1* is not found within *string-2*, #VALUE! is returned.
- *start-pos* designates a position outside *string-2*, #VALUE! is returned.

[*Example*:

FIND("de","abcdef") results in 4
FIND(A10,B10) results in 4, when A10 contains de, and B10 contains abcdef

end example]

18.17.7.120 FINDB

Syntax:

```
FINDB ( string-1 , string-2 , [ start-pos ] )
```

Description: Performs a case-sensitive search using a lexical comparison for the first occurrence of *string-1* in *string-2*, starting at byte position *start-pos* within *string-2*. (FINDB is intended for use with languages that use

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

the double-byte character set (DBCS), whereas FIND (§18.17.7.119) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

Name	Type	Description
string-1	text	Designate the string to be searched for within the string designated by *string-2*.
string-2	text	
start-pos	number	The number of the start position within *string-2* for which *string-1* is to be searched. The start position of the first byte is 1. If omitted, a position of 1 shall be assumed. *start-pos* shall be at least 0.

Return Type and Value: number – The start position of the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. If *string-1* is an empty string, it shall always be found in any *string-2* at position *start-pos*, or at position 1 if *start-pos* is omitted.

However, if

- *string-1* is not found within *string-2*, #VALUE! is returned.
- *start-pos* designates a position outside *string-2*, #VALUE! is returned.

[*Example*: Assuming 1-byte characters

`FINDB("de","abcdef")` results in 4
`FINDB(A10,B10)` results in 4, when `A10` contains `de`, and `B10` contains `abcdef`

end example]

18.17.7.121 FINV

Syntax:

`FINV` (*probability* , *degrees-freedom-1* , *degrees-freedom-2*)

Description: Computes the inverse of the F probability distribution. Given a value for *probability*, FINV seeks that value *x* such that `FDIST`(*x*, *degrees-freedom1*, *degrees-freedom2*) = *probability*. Thus, precision of FINV depends on precision of FDIST.

Arguments:

Name	Type	Description
probability	number	A probability associated with the F cumulative distribution.
degrees-	number	The number of degrees of freedom for the numerator,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
freedom-1		truncated to an integer.
degrees-freedom-2	number	The number of degrees of freedom for the denominator, truncated to an integer.

Return Type and Value: number – The inverse of the F probability distribution.

However, if

- *probability* < 0 or *probability* > 1, `#NUM!` is returned.
- *degrees-freedom-1* < 1 or *degrees-freedom-1* ≥ 10^{10}, `#NUM!` is returned.
- *degrees-freedom-2* < 1 or *degrees-freedom-2* ≥ 10^{10}, `#NUM!` is returned.
- the implementation determines that a return value cannot be computed, `#N/A` is returned

[*Example*:

`FINV(0.5,3,4)` results in `0.940534076`

end example]

18.17.7.122 FISHER

Syntax:

```
FISHER ( x )
```

Description: Computes the Fisher transformation at *x*.

Mathematical Formula:

$$z' = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right)$$

where:

- *x* = argument *x*

Arguments:

Name	Type	Description
x	number	The number for which the transformation is wanted.

Return Type and Value: number – The Fisher transformation at *x*.

However, if

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- $x \leq -1$, `#NUM!` is returned.
- $x \geq 1$, `#NUM!` is returned.

[*Example*:

`FISHER(-0.43)` results in `-0.459896681`
`FISHER(0.578)` results in `0.659454094`

end example]

18.17.7.123 FISHERINV

Syntax:

`FISHERINV ( y )`

Description: Computes the inverse of the Fisher transformation.

Mathematical Formula:

$$x = \frac{e^{2y} - 1}{e^{2y} + 1}$$

where:

- y = argument y

Arguments:

Name	Type	Description
y	number	The number for which the inverse of the transformation is wanted.

Return Type and Value: number – The inverse of the Fisher transformation.

[*Example*:

`FISHERINV(-0.43)` results in `0.405321309`
`FISHERINV(0.578)` results in `0.521210269`

end example]

18.17.7.124 FIXED

Syntax:

`FIXED ( number [ , [ num-decimal ] [ , suppress-commas-flag ] ] )`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Description: Produces a string containing *number* rounded to *num-decimal* decimal places, using the same rounding algorithm as ROUND (§18.17.7.278). Thousands separator commas are included as determined by *suppress-commas-flag*.

Arguments:

Name	Type	Description
number	number	Designate the number that is to be formatted, truncated to integer.
num-decimal	number	Designate the number of decimal places to be used in the resulting string. If negative, *number* is rounded to the left of the decimal point. If omitted, a value of 2 shall be assumed.
suppress-commas-flag	logical	If TRUE, commas are not included; if FALSE or omitted, commas are included.

Return Type and Value: text – The string containing *number* rounded to *num-decimal* decimal places.

[*Example*:

`FIXED(1234567)` results in `1,234,567.00`
`FIXED(1234567.555555,4,TRUE)` results in `1234567.5556`
`FIXED(.555555,10)` results in `0.5555550000`
`FIXED(1234567,-3)` results in `1,235,000`

end example]

18.17.7.125 FLOOR

Syntax:

FLOOR (*x* , *significance*)

Description: Computes *x* rounded down, toward zero, to the nearest multiple of *significance*. Regardless of the sign of *x*, a value is rounded down when adjusted away from zero.

Arguments:

Name	Type	Description
x	number	The value to be rounded,
significance	number	The multiple to which *x* is to be rounded.

Return Type and Value: number – The rounded-down value of *x*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if *x* and *significance* have different signs, `#NUM!` is returned.

[*Example*:

`FLOOR(2.5,1)` rounds 2.5 down to nearest multiple of 1; that is, to 2
`FLOOR(-2.5,-2)` rounds -2.5 down to nearest multiple of -2; that is, to `-2`
`FLOOR(1.5,0.1)` rounds 1.5 down to the nearest multiple of 0.1; that is, to `1.5`
`FLOOR(0.234,0.01)` rounds 0.234 down to the nearest multiple of 0.01; that is, to 0.23

end example]

18.17.7.126 FORECAST

Syntax:

FORECAST (*x* , *known-ys* , *known-xs*)

Description: Calculates, or predicts, a future value by using existing values. The predicted value is a y-value for a given x-value. The known values are existing x-values and y-values, and the new value is predicted by using linear regression.

Mathematical Formula:

FORECAST=a+bx, where:

$a = \bar{y} - b\bar{x}$

and:

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

where:

- x = a sample value
- $\bar{x}$ is the sample mean AVERAGE(known-xs)
- y = a sample value
- $\bar{y}$ is the sample mean AVERAGE(known-ys)

Arguments:

Name	Type	Description
x	number	The data point for which a value is to be predicted.
known-xs	array, reference	The independent data.
known-ys	array, reference	The dependent data.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The future value.

However, if

- *known-xs* and *known-ys* are empty or contain a different number of data points, the return value is unspecified.
- The variance of *known-xs* equals zero, the return value is unspecified.

[*Example*:

`FORECAST(30,{6,7,9,15,21},{20,28,31,38,40})` results in `10.60725309`

end example]

18.17.7.127 FREQUENCY

Syntax:

FREQUENCY (*data-array* , *bins-array*)

Description: Calculates how often values occur within a range of values. A call to FREQUENCY shall be an array formula.

Arguments:

Name	Type	Description
data-array	array, reference to number	Set of values for which frequencies are to be computed. If *data-array* contains no values, FREQUENCY returns an array of zeros. Cells containing text or that are empty are ignored.
bins-array	array, reference	Set of intervals into which the values in *data-array* are to be grouped. If *bins-array* contains no values, FREQUENCY returns the number of elements in *data-array*.

Return Type and Value: vertical array of numbers – The frequency at which values occur within a range of values. The number of elements in the returned array is one more than the number of elements in *bins-array*. The extra element contains the count of any values above the highest interval.

[*Example*:

If the cells A2:A10 contain 79, 85, 78, 85, 50, 81, 95, 88, and 97, and the cells B2:B4 contain 70, 79, and 89, `FREQUENCY(A2:A10,B2:B4)` results in a vertical array containing 1 (50), 2 (79, 78), 4 (85, 85, 81, 88), and 2 (95, 97).

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.128 FTEST

Syntax:

FTEST (*array-1* , *array-2*)

Description: Computes the result of an F-test.

Arguments:

Name	Type	Description
array-1	number, name, array, reference to number	If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
array-2		

Return Type and Value: number – The two-tailed probability that the variances in *array-1* and *array-2* are not significantly different.

However, if

- The number of data points in *array-1* or *array-2* is less than 2, the return value is unspecified.
- The variance of *array-1* or *array-2* is zero, the return value is unspecified.

[*Example*:

If the cells D6:D10 contain 6, 7, 9, 15, and 21, and the cells E6:E10 contain 20, 28, 31, 38, and 40, `FTEST(D6:D10,E6:E10)` results in `0.648317847`

end example]

18.17.7.129 FV

Syntax:

FV (*rate* , *nper* , *pmt* [, [*pv*] [, [*type*]]])

Description: Computes the future value of an investment based on periodic, constant payments and a constant interest rate.

Arguments:

Name	Type	Description
rate	number	The interest rate.
nper	number	The total number of payment periods, truncated to integer.
pmt	number	The payment made each period; it cannot change over the life of the annuity. [*Note*: Typically, *pmt* contains

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		principal and interest, but no other fees or taxes. *end note*] If omitted, *pv* shall be provided.
pv	number	The the present value, or the lump-sum amount that a series of future payments is worth right now. If omitted, it is assumed to be 0, and *pmt* shall be provided.
type	number	The timing of the payment, truncated to integer, as follows: `0` Payment at the end of the period `1` Payment at the beginning of the period

Arguments representing cash paid by investor shall be expressed as negative numbers; arguments representing cash received by the investor shall be expressed as positive numbers.

Return Type and Value: number – The future value of an investment based on periodic, constant payments and a constant interest rate.

However, if *type* is any number other than 0 or 1, #NUM! is returned.

[*Example*:

```
FV(0.06/12,10,-200,-500,1)     2,581.40
FV(0.12/12,12,-1000) results in 12,682.50
FV(0.11/12,35,-2000,,1) results in 82,846.25
FV(0.06/12,12,-100,-1000,1) results in 2,301.40
```

end example]

18.17.7.130 FVSCHEDULE

Syntax:

FVSCHEDULE (*principal* , *schedule*)

Description: Computes the future value of an initial principal after applying a series of compound interest rates. [*Note*: This function can be used to calculate the future value of an investment with a variable or adjustable rate. *end note*]

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
principal	number	The present value.
schedule	array	Set of interest rates to apply. The values in this array can be numbers or blank cells. Blank cells are taken as zeros (i.e., no interest).

Return Type and Value: number – The future value of an initial principal after applying a series of compound interest rates.

However, if any element of the array *schedule* is not a number and not blank, #VALUE! is returned.

[*Example*:

`FVSCHEDULE(1,{0.09,0.11,0.1})` results in `1.33089`

end example]

18.17.7.131 GAMMADIST

Syntax:

`GAMMADIST ( x , alpha , beta , cumulative-flag )`

Description: Computes the gamma distribution.

Mathematical Formula:

The equation for the gamma probability density function is:

$$f(x,\alpha,\beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}$$

The standard gamma probability density function is:

$$f(x,\alpha) = \frac{x^{\alpha-1}e^{-x}}{\Gamma(\alpha)}$$

When *alpha* = 1, `GAMMADIST` returns the exponential distribution with:

$$\lambda = \frac{1}{\beta}$$

where:

- α = argument *alpha*
- β = argument *beta*
- x = argument *x*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

For a positive integer n, when *alpha* = n/2, *beta* = 2, and cumulative = TRUE, GAMMADIST returns (1-CHIDIST(x)) with n degrees of freedom.

When *alpha* is a positive integer, GAMMADIST is also known as the Erlang distribution.

Arguments:

Name	Type	Description
x	number	The value at which the distribution is to be evaluated.
alpha	number	A parameter of the distribution.
beta	number	A parameter of the distribution. If *beta* = 1, GAMMADIST returns the standard gamma distribution.
cumulative-flag	logical	Determines the form of the function. If TRUE, GAMMADIST returns the cumulative distribution function; if FALSE, it returns the probability density function.

Return Type and Value: number – The gamma distribution.

However, if

- *x* < 0, #NUM! is returned.
- *alpha* ≤ 0 or *beta* ≤ 0, #NUM! is returned.

[*Example*:

GAMMADIST(10,9,2,FALSE) results in 0.03263902
GAMMADIST(10,9,2,TRUE) results in 0.068093631

end example]

18.17.7.132 GAMMAINV

Syntax:

GAMMAINV (*probability* , *alpha* , *beta*)

Description: Computes the inverse of the gamma distribution. Given a value for *probability*, GAMMAINV seeks that value *x* such that GAMMADIST(*x*, *alpha*, *beta*, TRUE) = *probability*. Thus, the precision of GAMMAINV depends on the precision of GAMMADIST.

Arguments:

Name	Type	Description
probability	number	The probability associated with the gamma distribution.
alpha	number	A parameter of the distribution.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
beta	number	A parameter of the distribution. If *beta* = 1, GAMMAINV returns the standard gamma distribution.

Return Type and Value: number – The inverse of the gamma distribution.

However, if

- *probability* < 0 or *probability* > 1, `#NUM!` is returned.
- *alpha* ≤ 0 or *beta* ≤ 0, `#NUM!` is returned.
- the implementation determines that a return value cannot be computed, `#N/A` is returned.

[*Example*:

`GAMMAINV(0.068,9,2)` results in `9.997130086`

end example]

18.17.7.133 GAMMALN

Syntax:

GAMMALN (*x*)

Description: Computes the natural logarithm of the gamma function.

Mathematical Formula:

$$GAMMALN = LN(\Gamma(x))$$

where:

$$\Gamma(x) = \int_0^\infty e^{-u} u^{x-1}\, du$$

where:

- x = argument x

Arguments:

Name	Type	Description
x	number	The value for which the gamma function is to be calculated.

Return Type and Value: number – The natural logarithm of the gamma function.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

However, if $x \leq 0$, `#NUM!` is returned.

[*Example*:

`GAMMALN(4.5)` results in `2.453736571`

end example]

18.17.7.134 GCD

Syntax:

GCD (*argument-list*)

Description: Computes the greatest common divisor of the one or more numbers, designated by *arguments* in *argument-list*.

Arguments:

Name	Type	Description
argument-list	numbers	The *arguments* in *argument-list* designate the values. Each argument is truncated to an integer.

Return Type and Value: number – The greatest common divisor of one or more numbers.

However, if any *argument* is negative, `#NUM!` is returned.

[*Example*:

`GCD(5)` results in `5`
`GCD(5,2)` results in `1`
`GCD(100,50,28)` results in `2`
`GCD(24.5,36.3)` results in `12`
`GCD(7,1)` results in `1`
`GCD(5,0)` results in `5`

end example]

18.17.7.135 GEOMEAN

Syntax:

GEOMEAN (*argument-list*)

Description: Computes the geometric mean of an array or range of positive data.

Mathematical Formula:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

$GM_{\bar{y}} = \sqrt[n]{y_1 y_2 y_3 \ldots y_n}$

where:

- n is the number of elements in *argument-list*.
- $y_1, y_2, y_3, \cdots, y_n$ are the values of the n-th element in *argument-list*.

Arguments:

Name	Type	Description
argument-list	logical, number, name, array, or reference to number.	The *argument*s in *argument-list* designate the values to be averaged. Logical values and text representations of numbers that entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The geometric mean of an array or range of positive data.

However, if the value of any data point ≤ 0, `#NUM!` is returned.

[*Example*:

`GEOMEAN(10.5,5.3,2.9)` results in `5.444454702`
`GEOMEAN(10.5,{5.3,2.9},"12")` results in `6.633780588`

end example]

18.17.7.136 GESTEP

Syntax:

`GESTEP ( number [ , step ] )`

Description: Tests if the value of *number* is greater than or equal to that of *step*.

Arguments:

Name	Type	Description
number	number	*number* is the value to test against *step*. If *step* is omitted, zero is used.
step	number	

Return Type and Value: number – 1 if *number* ≥ *step*; otherwise, 0.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*:

`GESTEP(5.6,-4.3)` results in `1`
`GESTEP(5.6,5.6)` results in `1`
`GESTEP(-5.6)` results in `0`

end example]

18.17.7.137 GETPIVOTDATA

Syntax:

```
GETPIVOTDATA ( data-field , pivot-table , field-1 , item-1
    [ , field-2 , item-2 [ , … ] ] )
```

Description: Retrieves data stored in a PivotTable report. Calculated fields or items and custom calculations are included in `GETPIVOTDATA` calculations.

Arguments:

Name	Type	Description
data-field	text	The name of the data field that contains the data to be retrieved.
pivot-table	reference to any cell, range of cells, or named range of cells in a PivotTable report	This information is used to determine which PivotTable report contains the data to be retrieved. If *pivot-table* is a range that includes two or more PivotTable reports, data shall be retrieved from whichever report was created most recently in the range.
field-1 through *field-n*	text	Argument pairs *field-1* and *item-1*, *field-2* and *item-2* through *field-n* and *item-n* are field names and item names that describe the data to be retrieved. The pairs can be in any order. Field names and names for items other than dates/times (which shall be expressed as numbers) and numbers shall be enclosed in quotation marks. For OLAP PivotTable reports, items can contain the source name of the dimension as well as the source name of the item. [*Example*: A field and item pair for an OLAP PivotTable might look like this: `"[Product]","[Product].[All Products].[Foods].[Baked Goods]"` *end example*] If the field and item arguments describe a single cell, the value of that cell is returned regardless of its value.
item-1 through *item-n*	text	

Return Type and Value: any – The data stored in a PivotTable report.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

However, if

- *pivot-table* is not a range in which a PivotTable report is found, the return value is unspecified.
- The arguments do not describe a visible field, the return value is unspecified.
- The arguments include a page field that is not displayed, the return value is unspecified.

[*Example*: Given the following data:

	A	B	C	D	E
2	Region	North			
3					
4	Sum of Sales		Product		
5	Month	Salesperson	Beverages	Produce	Grand Total
6	March	Buchanan	$ 3,522	$ 10,201	$ 13,723
7		Davolio	$ 8,725	$ 7,889	$ 16,614
8	March Total		$ 12,247	$ 18,090	$ 30,337
9	April	Buchanan	$ 5,594	$ 7,265	$ 12,859
10		Davolio	$ 5,461	$ 668	$ 6,129
11	April Total		$ 11,055	$ 7,933	$ 18,988
12	Grand Total		$ 23,302	$ 26,023	$ 49,325

`GETPIVOTDATA("Sales",$A$4)` returns the grand total of the Sales field, `$49,325`.

`GETPIVOTDATA("Sum of Sales",$A$4)` also returns the grand total of the Sales field, `$49,325`; the field name can be entered exactly as it looks on the sheet, or as its root (without "Sum of," "Count of," and so forth).

`GETPIVOTDATA("Sales",$A$4,"Month","March")` returns the grand total for March, `$30,337`.

`GETPIVOTDATA("Sales",$A$4,"Month","March","Product","Produce","Salesperson","Buchanan")` returns `$10,201`.

`GETPIVOTDATA("Sales",$A$4,"Region","South")` is unspecified because the South region data is not visible.

`GETPIVOTDATA("Sales",$A$4,"Product","Beverages","Salesperson","Davolio")` is unspecified because there is no total value of beverage sales for Davolio.

end example]

18.17.7.138 GROWTH

Syntax:

GROWTH (*known-ys* [, [*known-xs*] [, [*new-xs*] [, *const-flag*]])

Description: Computes predicted exponential growth by using existing data. GROWTH can also fit an exponential curve to existing x-values and y-values.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
known-ys	array	Set of y-values already known in the relationship y=b*m^x. If the array *known-ys* is a single column, then each column of *known-xs* is interpreted as a separate variable. If the array *known-ys* is a single row, then each row of *known-xs* is interpreted as a separate variable.
known-xs	array	Set of x-values that might already be know in the relationship y=b*m^x. The array *known-xs* can include one or more sets of variables. If only one variable is used, *known-ys* and *known-xs* can be ranges of any shape, as long as they have equal dimensions. If more than one variable is used, *known-ys* shall be a vector (that is, a *known-ys* with a height of one row or a width of one column). If *known-xs* is omitted, it is assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
new-xs	array	A set of new x-values for which GROWTH is to return corresponding y-values. *new-xs* shall include a column (or row) for each independent variable, just as *known-xs* does. So, if *known-ys* is in a single column, *known-xs* and *new-xs* shall have the same number of columns. If *known-ys* is in a single row, *known-xs* and *new-xs* shall have the same number of rows. If *new-xs* are omitted, it is assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
const-flag	logical	Specifies whether to force the constant b to equal 1. If TRUE or omitted, *b* is calculated normally. If FALSE, *b* is set equal to 1 and the m-values are adjusted so that y= m^x.

Return Type and Value: array – The y-values for a series of new x-values.

However, if any of the numbers in *known-ys* are zero or negative, `#NUM!` is returned.

[*Example*: Given the following data:

	A	B	C
1	Month	Units	Formula (corresponding units)
2	11	33,100	32618.20377
3	12	47,300	47729.42261
4	13	69,000	69841.30086
5	14	102,000	102197.0734
6	15	150,000	149542.4867

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	A	B	C
7	16	220,000	218821.8762
8	Month	Formula (Predicted Units)	
9	17	320,196.72	
10	18	468,536.05	

When `GROWTH(A2:B4,A6:B8)` is an array formula spanning cells C2:C7, those cells take on the results shown. When `GROWTH(A2:B4,A6:B8,A9:A10)` is an array formula spanning cells B9:B10, those cells take on the results shown.

end example]

18.17.7.139 HARMEAN

Syntax:

HARMEAN (*argument-list*)

Description: Computes the harmonic mean of a data set.

Mathematical Formula:

$$\frac{1}{H} = \frac{1}{n}\sum\frac{1}{Y_i}$$

where:

- n = number of elements in *argument-list*
- Y_i = each element in *argument-list*

Arguments:

Name	Type	Description
argument-list	logical, text, number, name, array, or reference to number.	The *arguments* in *argument-list* designate the values to be averaged. Argument values can be numbers, or names, arrays, or references that contain numbers. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The harmonic mean of a data set.

However, if the value of any data point ≤ 0, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

`HARMEAN(4.6,5.8,8.3,7)` results in `6.124222`
`HARMMEAN(10.5,{5.3,2.9},"12")` results in `5.617360`

end example]

18.17.7.140 HEX2BIN

Syntax:

`HEX2BIN ( number [ , num-bin-digits ] )`

Description: Makes the binary equivalent of *number*, with the result having *num-bin-digits* digits.

Arguments:

Name	Type	Description
number	text	A 10-digit hexadecimal number in a string that is to be converted to a binary string. *number* is not case-sensitive. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (40th bit from the right) representing the sign bit.
num-bin-digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-bin-digits* is ignored and the result has 10 digits. If *num-bin-digits* is omitted, the minimum number of digits is used in the result. *num-bin-digits* is truncated to an integer.

Return Type and Value: text – The binary equivalent of *number*.

However, if

- *number* is outside the range `"FFFFFFFE00"` (1111111111111111111111111111111000000000 binary, -512 decimal) to `"1FF"` (0000000000000000000000000000000111111111 binary, 511 decimal), inclusive, `#NUM!` is returned.
- *number* contains one or more non-hexadecimal digits, `#NUM!` is returned.
- *number* contains more than 10 hexadecimal digits, `#NUM!` is returned.
- *number* needs more digits that *num-bin-digits*, `#NUM!` is returned.
- *num-bin-digits* is negative or > 10, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

HEX2BIN("fE") results in 11111110
HEX2BIN("FFFFFFFFFE") results in 1111111110
HEX2BIN("2") results in 10
HEX2BIN("F",6) results in 001111

end example]

18.17.7.141 HEX2DEC

Syntax:

HEX2DEC (*number*)

Description: Makes the decimal equivalent of *number*.

Arguments:

Name	Type	Description
number	string	A 10-digit hexadecimal number in a string that is to be converted to a decimal number. *number* is not case-sensitive. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (40th bit from the right) representing the sign bit.

Return Type and Value: number – The decimal equivalent of *number*.

However, if

- *number* contains one or more non-hexadecimal digits, #NUM! is returned.
- *number* contains more than 10 hexadecimal digits; that is, *number* is outside the range "8000000000" (-548,755,813,888 decimal) to "7FFFFFFFF" (548,755,813,887 decimal), inclusive, #NUM! is returned.

[*Example*:

HEX2DEC("fE") results in 254
HEX2DEC("FFFFFFFFFE") results in -2
HEX2DEC("F000000000") results in -68719476736

end example]

18.17.7.142 HEX2OCT

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

HEX2OCT (*number* [, *num-oct-digits*])

Description: Makes the octal equivalent of *number*, with the result having *num-oct-digits* digits.

Arguments:

Name	Type	Description
number	text	A 10-digit hexadecimal number in a string that is to be converted to an octal string. *number* is not case-sensitive. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use two-complement representation with the left-most bit (40th bit from the right) representing the sign bit.
num-oct-digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-oct-digits* is ignored and the result has 10 digits. If *num-oct-digits* is omitted, the minimum number of digits is used in the result. *num-oct-digits* is truncated to an integer.

Return Type and Value: text – The octal equivalent of *number*.

However, if

- *number* is outside the range `"FFE0000000"` (17774000000000 octal, -536,870,912 decimal) to `"1FFFFFFF"` (00003777777777 octal, 536,870,911 decimal), inclusive, `#NUM!` is returned.
- *number* contains one or more non-hexadecimal digits, `#NUM!` is returned.
- *number* contains more than 10 hexadecimal digits, `#NUM!` is returned.
- *number* needs more digits that *num-oct-digits*, `#NUM!` is returned.
- *num-oct-digits* is negative or > 10, `#NUM!` is returned.

[*Example*:

`HEX2OCT("fE")` results in `376`
`HEX2OCT("FFFFFFFFFE")` results in `7777777776`
`HEX2OCT("2")` results in `2`
`HEX2OCT("F",6)` results in `000017`

end example]

18.17.7.143 HLOOKUP

Syntax:

HLOOKUP (*lookup-value* , *table-array* , *row-index-num* [, [*range-lookup-flag*]])

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Performs a horizontal search for a value in the top row of a table or an array, noting the column in which the matching value is found. From that column, the value from a given row is returned.

Arguments:

Name	Type	Description
lookup-value	value or reference.	The value to be located in the first row of the table. If *range-lookup* is FALSE and *lookup-value* is a string, the wildcard characters, question mark (?) and asterisk (*), can be included in *lookup-value*. A question mark matches any single character; an asterisk matches any sequence of characters. To find a question mark or asterisk, use a tilde (~) before the character.
table-array	array, reference, name	Designates the table of information to be searched. The values in the first row of *table-array* can be text, numbers, or logical values. If *range-lookup-flag* is TRUE, the values in the first row of *table-array* shall be placed in "ascending order", as follows: …, -2, -1, 0, 1, 2, …, A–Z, FALSE, TRUE. If *range-lookup-flag* is FALSE, *table-array*'s values need not be sorted. Uppercase and lowercase text is treated as equivalent.
row-index-num	number	The row number in *table-array* from which the matching value is to be returned. (A *row-index-num* of 1 returns the first row value in *table-array*, a *row-index-num* of 2 returns the second row value in *table-array*, and so on.)
range-lookup-flag	logical	Specifies whether HLOOKUP is to find an exact or approximate match. If TRUE or omitted, an approximate match is returned. That is, if an exact match is not found, the next largest value that is less than *lookup-value* is returned. If FALSE, an exact match is performed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: any – The value from a given row number, where the column is determined by a search of the top row looking for a match with a given value.

However, if

- An exact match is performed, but no match is found, #N/A is returned.
- *row-index-num* is less than 1, #VALUE! is returned.
- *row-index-num* is greater than the number of rows in *table-array*, #REF! is returned.
- *lookup-value* is smaller than the smallest value in the first row of *table-array*, #N/A is returned.

[*Example*: Given the following data:

©ISO/IEC 2016 – All rights reserved

	A	B	C
1	Axles	Bearings	Bolts
2	4	6	9
3	5	7	10
4	6	8	11

`HLOOKUP("Axles",A1:C4,2,TRUE)` results in 4
`HLOOKUP("Bearings",A1:C4,3,FALSE)` results in 7
`HLOOKUP("B",A1:C4,3,TRUE)` results in 5
`HLOOKUP("Bolts",A1:C4,4)` results in 11
`HLOOKUP(3,{1,2,3;"a","b","c";"d","e","f"},2,TRUE)` results in c

end example]

18.17.7.144 HOUR

Syntax:

`HOUR` (*time-value*)

Description: Computes the hour for the date and/or time having the given *time-value*.

Arguments:

Name	Type	Description
time-value	number	The date and/or time whose hour is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its integer part is ignored, or as a *string-constant* having any date and/or time format, in which case, any date information shall be ignored.

Return Type and Value: number – The hour for the date and/or time having the given *time-value*. The returned value shall be in the range 0–23.

However, if *time-value* is out of range for the current date system, `#NUM!` is returned.

[*Example*:

`HOUR(DATE(2006,2,26)+TIME(2,10,20))` results in 2
`HOUR(TIME(22,56,34))` results in 22
`HOUR(0)` results in 0, since serial date-time 0 represents `00:00:00`
`HOUR(10.5)` results in 12, since serial date-time .5 represents `12:00:00`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

HOUR("22-Oct-2001 10:53:12") results in 10
HOUR("10:53:12 pm") results in 22
HOUR("22:53:12") results in 22

end example]

18.17.7.145 HYPERLINK

Syntax:

HYPERLINK (*link-location* [, [*friendly-name*]])

Description: Creates a shortcut that opens a document stored on a network server, an intranet, or the Internet. When the cell that contains the HYPERLINK function call is clicked, the file stored at *link-location* is opened.

Arguments:

Name	Type	Description
link-location	text	The location to be opened as text. *link-location* can refer to a place in a document—such as a specific cell or named range in a SpreadsheetML worksheet or workbook, or to a bookmark in a WordprocessingML document. The *link-location* can be to a location on a local hard disk drive, the path to a location on a server or a Uniform Resource Locator (URL) to a location on the Internet or an intranet. [*Note*: Using a URL is the preferred way of specifying *link-location*. *end note*] If the location specified in *link-location* does not exist or cannot be navigated, an unspecified error is produced when the cell is clicked. *link-location* can be a string or a reference to a cell containing a string.
friendly-name	text, number, name	The value that is displayed in the cell. If omitted, the cell displays *link-location*. *friendly-name* can be a value, a text string, a name, or a cell that contains the jump text or value. If the evaluation of *friendly-name* results in an error value, the cell displays that error value rather than the jump text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: text – The value of *friendly-name*, if it is specified; otherwise, the value of *link-location*.

[*Example*:

HYPERLINK("http://example.openxmlformats.org/report/budget report.xls","Click for report"), which opens a worksheet named "budget report.xls" that is stored on the Internet at the location example.openxmlformats.org/report, and displays the text "Click for report".

©ISO/IEC 2016 – All rights reserved

HYPERLINK("D:\FINANCE\1stqtr.xls",H10), which opens the file 1stqtr.xls that is stored in a directory named Finance on drive D, and displays the numeric value stored in cell H10.

end example]

18.17.7.146 HYPGEOMDIST

Syntax:

HYPGEOMDIST (*sample-successes* , *number-sample* , *population-successes* , *number-population*)

Description: Computes the hypergeometric distribution; that is, the probability of a given number of sample successes, given the sample size, population successes, and population size.

Mathematical Formula:

$$P(X = x) = h(x, n, M, N) = \frac{\binom{M}{x}\binom{N-M}{n-x}}{\binom{N}{n}}$$

where:

- M = *population-successes* argument
- N = *number-population* argument
- x = *sample-successes* argument
- n = *number-sample* argument

Arguments:

Name	Type	Description
sample-successes	number	The number of successes in the sample, truncated to integer.
number-sample	number	The size of the sample, truncated to integer.
population-successes	number	The number of successes in the population, truncated to integer.
number-population	number	The population size, truncated to integer.

Return Type and Value: number – The hypergeometric distribution.

However, if

- *sample-successes* < 0 or *sample-successes* is greater than the lesser of *number-sample* and *population-successes*, #NUM! is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- *sample-successes* is less than the larger of 0 or (*number-sample* - *number-population* + *population-successes*), `#NUM!` is returned.
- *number-sample* ≤ 0 or *number-sample* > *number-population*, `#NUM!` is returned.
- *population-successes* ≤ 0 or *population-successes* > *number-population*, `#NUM!` is returned.
- *number-population* ≤ 0, `#NUM!` is returned.

[*Example*:

`HYPGEOMDIST(1,4,8,20)` results in `0.363261`

end example]

18.17.7.147 IF

Syntax:

`IF (` *logical-value* `, [` *value-if-true* `][ , [` *value-if-false* `]] )`

Description: Tests *logical-value*, and if it is TRUE, *value-if-true* is evaluated and returned; otherwise, *value-if-false* is evaluated and returned.

Arguments:

Name	Type	Description
logical-value	logical	The value to be tested.
value-if-true	any	The value returned if *logical-value* is TRUE. If *logical-value* is TRUE and *value-if-true* is omitted, this argument evaluates to 0. *value-if-true* can contain up to seven levels of nested `IF` function calls. [*Note*: *value-if-true* and *value-if-false* need not evaluate to results of the same type. *end note*]
value-if-false	any	The value returned if *logical-value* is FALSE. If *logical-value* is FALSE and *value-if-false* and its preceding comma is omitted, this argument evaluates to FALSE. If *logical-value* is FALSE and *value-if-false* is omitted, but its preceding comma is present, this argument evaluates to 0. *value-if-false* can contain at least seven levels of nested `IF` function calls. [*Note*: *value-if-true* and *value-if-false* need not evaluate to results of the same type. *end note*]

If any argument is an array, every element of that array shall be evaluated when that argument is evaluated.

Return Type and Value: any – *value-if-true*, if *logical-value* is TRUE; otherwise, *value-if-false*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

`IF(10>5,"Yes","No")` results in `Yes`
`IF(10>5,"Yes")` results in `Yes`
`IF(10>5,"Yes",)` results in `Yes`
`IF(10<5,"Yes")` results in `FALSE`
`IF(10<5,"Yes",)` results in `0`

`IF(10>5,,"No")` results in `0`
`IF(10>5,,)` results in `0`
`IF(10>5,"Yes",20)` results in `Yes`
`IF(10<5,"Yes",20)` results in `20`

end example]

18.17.7.148 IFERROR

Syntax:

`IFERROR ( value , value-if-error )`

Description: Provides a simpler and more efficient way of trapping and handling errors. It allows the generation of user-defined error text for a function call that can result in an error.

Arguments:

Name	Type	Description
value	any	The value that is checked for an error (i.e., any of the following: `#N/A`, `#VALUE!`, `#REF!`, `#DIV/0!`, `#NUM!`, `#NAME?`, or `#NULL!`). If *value* is an empty cell, it is treated as an empty string.
value-if-error	any	The value to return if *value* evaluates to an error. If *value-if-error* is an empty cell, it is treated as an empty string.

Return Type and Value: any – *value*, if *value* is not an error; otherwise, *value-if-error*. If *value* is an array formula, an array of results for each cell in the range specified in value, is returned.

[*Example*: Consider the case in which A3 contains 55, and B3 contains 0:

`A3/B3` results in `#DIV/0`

`IFERROR(A3/B3,"Error in calculation")` results in `Error in calculation`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.149 IMABS

Syntax:

`IMABS ( complex-number )`

Description: Computes the absolute value of *complex-number*.

Mathematical Formula:

$IMABS(z) = |z| = \sqrt{x^2 + y^2}$

where:

- z = argument *complex-number*, expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the absolute value is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: number – The absolute value of *complex-number*.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMABS("3+4i")` results in 5
`IMABS("-2.5-34.6j")` results in 34.69020035

end example]

18.17.7.150 IMAGINARY

Syntax:

`IMAGINARY ( complex-number )`

Description: Computes the imaginary coefficient of *complex-number*.

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the imaginary coefficient is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The imaginary coefficient of *complex-number*.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMAGINARY("3+4i")` results in 4
`IMAGINARY("-2.5-34.6j")` results in `34.6`

end example]

18.17.7.151 IMARGUMENT

Syntax:

`IMARGUMENT` (*complex-number*)

Description: Computes the argument θ, an angle expressed in radians, such that for a complex number *complex-number* having the form *x+yi*:

$x + yi = |x + yi| \times e^{\theta} = |x + yi|(\cos\theta + i\sin\theta)$

Mathematical Formula:

$$IMARGUMENT(z) = \tan^{-1}\left(\frac{y}{x}\right) = \theta$$

where:

- $\theta \in (-\pi; \pi]$
- z = argument *complex-number*, expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number in *x + yi* or *x + yj* text format.

Return Type and Value: number – The angle θ, expressed in radians.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMARGUMENT("13+4i")` results in `0.298498932`
`IMARGUMENT("-2.5-5j")` results in `-2.034443936`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.152 IMCONJUGATE

Syntax:

IMCONJUGATE (*complex-number*)

Description: Computes the complex conjugate of the complex number *complex-number*.

Mathematical Formula:

$ICONJUGATE(x + yi) = \bar{z} = (x - yi)$

where:

- $\bar{z}$ = conjugate of the argument *complex-number*, expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the complex conjugate is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the complex conjugate of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMCONJUGATE("2.3+4.5i")` results in `2.3-4.5i`
`IMCONJUGATE("-1-4j")` results in `-1+4j`

end example]

18.17.7.153 IMCOS

Syntax:

IMCOS (*complex-number*)

Description: Computes the cosine of the complex number *complex-number*.

Mathematical Formula:

$\cos(x + yi) = \cos(x)\cosh(y) - \sin(x)\sinh(y)i$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

where:

- the argument *complex-number is* expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the cosine is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the cosine of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMCOS("2.3+4.5i")` results in `-29.9918288739746-33.5589799796873i`
`IMCOS("-1-4j")` results in `14.7547011704838-22.963673499193j`

end example]

18.17.7.154 IMDIV

Syntax:

`IMDIV (` *complex-number-1* `,` *complex-number-2* `)`

Description: Computes the quotient from dividing two complex numbers.

Mathematical Formula:

$$IMDIV(z_1, z_2) = \frac{(a+bi)}{(c+di)} = \frac{(ac+bd)+(bc-ad)i}{c^2+d^2}$$

where:

- z_1 = argument *complex-number-1* expressed in the form of a + bi
- z_2 = argument *complex-number-2* expressed in the form of c + di

Arguments:

Name	Type	Description
complex-number-1	text	Complex numbers in *x + yi* or *x + yj* text format; they designate the dividend and divisor, respectively.
complex-number-2	text	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: text – A string containing the quotient from *complex-number-1* / *complex-number-2*, in *x+yi* or *x+yj* text format.

However, if

- *complex-number-2* is zero, `#NUM!` is returned.
- *complex-number-1* or *complex-number-2* is ill-formed, #NUM! is returned.

[*Example*:

`IMDIV("13+4i","5+3i")` results in `2.26470588235294-0.558823529411765i`
`IMDIV("-3-3.5i","5+3i")` results in `-0.75-0.25i`

end example]

18.17.7.155 IMEXP

Syntax:

`IMEXP` (*complex-number*)

Description: Computes the exponential of the complex number *complex-number*.

Mathematical Formula:

$$IMEXP(\ \ ) = e^{(x+yi)} = e^x e^{yi} = e^x(\cos y + i \sin y)$$

where:

- z = the argument *complex-number*, expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the exponential is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the exponential of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

`IMEXP("2.3+4.5i")` results in `-2.10251576423113-9.750063748668l8i`
`IMEXP("-1-4j")` results in `-0.240462049968584+0.278412079051034j`

end example]

18.17.7.156 IMLN

Syntax:

`IMLN ( complex-number )`

Description: Computes the natural logarithm of *complex-number*.

Mathematical Formula:

$$\ln(x + yi) = \ln\sqrt{x^2 + y^2} + i\tan^{-1}\left(\frac{y}{x}\right)$$

where:

- the argument *complex-number* is expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the natural logarithm is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – The natural logarithm of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed or zero, `#NUM!` is returned.

[*Example*:

`IMLN("3+4i")` results in `1.6094379124341+0.927295218001612i`
`IMLN("-2.5-34.6j")` results in `3.54645723627033-1.64292531532225j`

end example]

18.17.7.157 IMLOG10

Syntax:

`IMLOG10 ( complex-number )`

Description: Computes the base-10 logarithm of *complex-number*.

Mathematical Formula:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The common logarithm of a complex number can be calculated from the natural logarithm as follows:

$$\log_{10}(x + yi) = (\log_{10} e) \ln(x + yi)$$

where:

- the argument *complex-number* is expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the base-10 logarithm is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – The base-10 logarithm of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed or zero, `#NUM!` is returned.

[*Example*:

`IMLOG10("3+4i")` results in `10.698970004336019+0.4027191962733 73i`
`IMLOG10("-2.5-34.6j")` results in `11.54020680801806-0.713513398623614j`

end example]

18.17.7.158 IMLOG2

Syntax:

`IMLOG2 ( complex-number )`

Description: Computes the base-2 logarithm of *complex-number*.

Mathematical Formula:

The base-2 logarithm of a complex number can be calculated from the natural logarithm as follows:

$$\log_2(x + yi) = (\log_2 e) \ln(x + yi)$$

where:

- the argument *complex-number* is expressed in the form x + yi

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
complex-number	text	The complex number for which the base-2 logarithm is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – The base-2 logarithm of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed or zero, `#NUM!` is returned.

[*Example*:

IMLOG2("3+4i") results in `2.32192809506607+1.33780421255394i`
IMLOG2("-2.5-34.6j") results in `5.11645626788577-2.37024020514877j`

end example]

18.17.7.159 IMPOWER

Syntax:

`IMPOWER ( complex-number , y )`

Description: Computes the complex number *complex-number* raised to the power *y*.

Mathematical Formula:

$(x + yi)^n = r^n e^{in\theta} = r^n \cos\theta + ir^n \sin n$

where:

$r = \sqrt{x^2 + y^2}$

and:

$= \tan^{-1}\left(\frac{y}{x}\right)$

and:

$\theta \in (-\pi; \pi]$

where:

- argument *complex-number* is expressed in the form x + yi
- *n* = argument *y*

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
complex-number	text	The complex number in *x + yi* or *x + yj* text format.
y	number	The exponent to which *complex-number* is to be raised.

Return Type and Value: text – A string containing *complex-number*^{y}, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed or zero, `#NUM!` is returned.

[*Example*:

`IMPOWER("2.3+4.5i",2.5)` results in `-52.9752689709953+22.138528463954i`
`IMPOWER("-1-4j",-3.56)` results in `6.34818926783845E-003+1.16156377299512E-003j`

end example]

18.17.7.160 IMPRODUCT

Syntax:

`IMPRODUCT (` *argument-list* `)`

Description: Multiplies the values of its complex number arguments.

Mathematical Formula:

$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$

where:

- Each element of *argument-list* is a complex number expressed in the form x + yi

Arguments:

Name	Type	Description
argument-list	text	Each *argument* in *argument-list* is a complex number string in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the product of the values of its arguments, in *x+yi* or *x+yj* text format.

However, if any *argument* in *argument-list* is ill-formed, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

`IMPRODUCT("13+4i")` results in `13+4i`
`IMPRODUCT("-3-3.5i","5+3i")` results in `-4.5-26.5i`
`IMPRODUCT("1.3-2j","-3.4+3j","2.3-6j")` results in `67.834+15.13j`

end example]

18.17.7.161 IMREAL

Syntax:

`IMREAL ( complex-number )`

Description: Computes the real coefficient of *complex-number*.

where:

- argument *complex-number* is expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the real coefficient is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: number – The real coefficient of *complex-number*.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMREAL("3+4i")` results in 3
`IMREAL("-2.5-34.6j")` results in `-2.5`

end example]

18.17.7.162 IMSIN

Syntax:

`IMSIN ( complex-number )`

Description: Computes the sine of the complex number *complex-number*.

Mathematical Formula:

$$\sin(x + yi) = \sin(x)\cosh(y) - \cos(x)\sinh(y)i$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

where:

- argument *complex-number* is expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the sine is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the sine of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMSIN("2.3+4.5i")` results in `33.567264016308-29.9844272159606i`
`IMSIN("-1-4j")` results in `-22.9790855778861-14.7448051885587j`

end example]

18.17.7.163 IMSQRT

Syntax:

`IMSQRT ( complex-number )`

Description: Computes the square root of the complex number *complex-number*.

Mathematical Formula:

$$\sqrt{x+yi} = \sqrt{r}\cos\left(\frac{\theta}{2}\right) + i\sqrt{r}\sin\left(\frac{\theta}{2}\right)$$

where:

$$r = \sqrt{x^2+y^2}$$

and:

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

and:

$$\theta \in (-\pi;\pi]$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

where:

- argument *complex-number* is expressed in the form x + yi

Arguments:

Name	Type	Description
complex-number	text	The complex number for which the square root is being computed. *complex-number* shall be in *x + yi* or *x + yj* text format.

Return Type and Value: text – A string containing the square root of *complex-number*, in *x+yi* or *x+yj* text format.

However, if *complex-number* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMSQRT("2.3+4.5i")` results in `1.91751290835255+1.17339496918073i`
`IMSQRT("-1-4j")` results in `1.24962106768765-1.60048518044024j`

end example]

18.17.7.164 IMSUB

Syntax:

`IMSUB ( complex-number-1 , complex-number-2 )`

Description: Computes the difference of two complex numbers.

Mathematical Formula:

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

where:

- argument *complex-number-1* is expressed in the form a + bi
- argument *complex-number-2* is expressed in the form c + di

Arguments:

Name	Type	Description
complex-number-1	text	Complex numbers in *x + yi* or *x + yj* text format; they designate the minuend and subtrahend, respectively.
complex-number-2	text	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: text – A string containing *number-1* - *number-2*, in *x+yi* or *x+yj* text format.

However, if *complex-number-1* or *complex-number-2* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMSUB("13+4i","5+3i")` results in `8+i`
`IMSUB("-3-3.5i","5+3i")` results in `-8-6.5i`

end example]

18.17.7.165 IMSUM

Syntax:

`IMSUM` (*argument-list*)

Description: Adds the values of its arguments.

Mathematical Formula:

$(a + bi) + (c + di) = (a + c) + (b + d)i$

where:

- Each element of *argument-list* is a complex number expressed in the form x + yi

Arguments:

Name	Type	Description
argument-list	text	Each *argument* in *argument-list* is a complex number string in *x + yi* or *x + yj* text format.

Return Type and Value: text – The sum of the values of its arguments, in *x+yi* or *x+yj* text format.

However, if any *argument* in *argument-list* is ill-formed, `#NUM!` is returned.

[*Example*:

`IMSUM("3+4i")` results in `3+4i`
`IMSUM("3+4i","5-3i")` results in `8+i`

end example]

18.17.7.166 INDEX

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

array form: `INDEX ( array , [ row-number ] [ , [ column-number ] ] )`

reference form: `INDEX ( reference [ , [ row-number ] [ , [ column-number ] [ , [ area-number ] ] ] ] )`

Description: Locates a value or the reference to a value from within a table or range. There are two forms of the INDEX function: the array form and the reference form.

Arguments:

Name	Type	Description
array	array, reference	Table or range to be searched. If *array* contains only one row, the corresponding *row-number* argument is optional. If *array* contains only one column, the corresponding *column-number* argument is optional.
reference	reference	A reference to one or more cell ranges. If each area (§18.17.2.3) in *reference* contains only one row, *row-number* is optional. If each area contains only one column, *column-number* is optional.
row-number	number	*row-number* indicates the row in *array* (or *reference*) from which to return a value (or reference). If *row-number* is omitted, *column-number* shall be present. *column-number* indicates the column in *array* (or *reference*) from which to return a value (or *reference*). If *column-number* is omitted, *row-number* shall be present. If both the *row-number* and *column-number* arguments are used, INDEX returns the value (or *reference*) in the cell at the intersection of *row-number* and *column-number*. If *array* has more than one row and more than one column, and only *row-number* or *column-number* is used, INDEX returns an array of the entire row or column in array. If *row-number* or *column-number*, but not both, is 0, INDEX returns the array of values for the entire column or row, respectively. In the reference form, if *row-number* and *column-number* are both omitted, INDEX returns the area in reference specified by *area-number*.
column-number	number	
area-number	number	Indicates a range in *reference* from which to return the intersection of *row-number* and *column-number*. The first area selected or entered is numbered 1, the second 2, and so on. If *area-number* is omitted, 1 is assumed. [*Example*: If *reference* describes the cells `(A1:B4, D1:E4, G1:H4)`, then area-number 1 is the range A1:B4, *area-number* 2 is the range D1:E4, and *area-number* 3 is the range G1:H4. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: various – For the array form, returns a single value, a whole row, or a whole column from a table or an array, depending on the presence and values of the row and column number indexes.

For the reference form, returns a single reference, a whole row, or a whole column from a reference, depending on the presence and values of the row and column number indexes, and the area number.

However, for the array form

- *row-number* is outside the bounds of *array*, `#REF!` is returned.
- *column-number* is outside the bounds of *array*, `#REF!` is returned.

For the reference form

- *row-number* is outside the bounds of *reference*, `#REF!` is returned.
- *column-number* is outside the bounds of *reference*, `#REF!` is returned.
- *area-number* is outside the bounds of *reference*, `#REF!` is returned.

[*Example*:

`INDEX({"Apples","Lemons";"Bananas","Pears"},2,2)` results in `Pears`
`INDEX({"Apples","Lemons";"Bananas","Pears"},2,1)` results in `Bananas`
`INDEX({"Apples","Lemons"},,2)` results in `Lemons`
`INDEX({"Apples";"Bananas"},1)` results in `Apples`

Given the following data:

	A	B	C
1	Fruit	Price	Count
2	Apples	0.69	40
3	Bananas	0.34	38
4	Lemons	0.55	15
5	Oranges	0.25	25
6	Pears	0.59	40
7	Almonds	2.8	10

`INDEX(A2:C7,2,3)` results in `38`
`INDEX((A2:C4,A6:C7),2,2,2)` results in `2.8`
`INDEX((A2:C4,A6:C7),2,2,1)` results in `0.34`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.167 INDIRECT

Syntax:

`INDIRECT ( ` *ref-text* ` [ , [ ` *A1-ref-style-flag* ` ] ] )`

Description: Locates the reference specified by *ref-text* and evaluates that reference to get to its underlying value. [*Note*: This function should be used when the reference to a cell within a formula is to be changed without changing the formula itself. *end note*]

Arguments:

Name	Type	Description
ref-text	An A1-style reference, an R1C1-style reference, a name defined as a reference, or a reference to a cell as a string.	If *ref-text* refers to another workbook (i.e., it's an external reference), that other workbook shall be open.
A1-ref-style-flag	logical	Specifies the kind of reference that is contained in the cell *ref-text*. If TRUE or omitted, *ref-text* is interpreted as an A1-style reference (§18.17.2.3.1); otherwise, *ref-text* is interpreted as an R1C1-style reference (§18.17.2.3.2).

Return Type and Value: any – The underlying value of the location referred to by *ref-text*.

However, if

- *ref-text* is not a cell reference, `#REF!` is returned.
- *ref-text* refers to another workbook yet that other workbook is not currently open, the return value is unspecified.
- *ref-text* is a name, then the content of the named expression can have implementation-defined constraints.

[*Example*:

Given the following data:

	A	B
1	Data	Data
2	B2	1.333
3	B3	45

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	A	B
4	George	10
5	5	62

where A2 contains a reference to B2, A3 contains a reference to B3, A4 contains the defined name George that refers to B4, and A5 contains the row number of B5:

`INDIRECT($A$2)` results in `1.333`
`INDIRECT($A$3)` results in `45`
`INDIRECT($A$4)` results in `10`
`INDIRECT("B"&$A$5)` results in `62`

`INDIRECT("R[-1]C",FALSE)` uses the cell in the previous row and current column.
end example]

18.17.7.168 INFO

Syntax:

`INFO ( category )`

Description: Retrieves the operating environment value that corresponds to *category*.

Arguments:

Name	Type	Description
category	text	The string designated by *category* is not case-sensitive. The permitted strings are shown in the table below.

category	Meaning	Result Type
`"directory"`	Path of the current directory or folder.	text
`"memavail"`	Amount of memory available, in bytes.	number
`"memused"`	Amount of memory being used for data.	number
`"numfile"`	Number of active worksheets in the open workbooks.	number
`"origin"`	The absolute cell reference of the top and leftmost cell visible in the window, based on the current scrolling position, prefixed with `"$A:"`. [*Example*: Using cell D9 as an example, the return value would be $A:$D$9. *end example*]	text
`"osversion"`	Current operating system version.	text
`"recalc"`	Current recalculation mode: `"Automatic"` or `"Manual"`	text
`"release"`	Version of the implementation.	text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

category	Meaning	Result Type
"system"	Name of the operating environment.	text
"totmem"	Total memory available, including memory already in use, in bytes.	number

Return Type and Value: text – The operating environment value that corresponds to *category*.

However, if *category* is not one of the values defined above, #VALUE! is returned.

[*Example*:

INFO("directory") might result in e:\My Documents\
INFO(A10) might result in e:\My Documents\, where A10 contains directory
INFO("memavail") might result in 1048576

INFO("memused") might result in 1474464
INFO("numfile") might result in 5
INFO("origin") might result in $A:$C$536

INFO("osversion") might result in Windows (32-bit) NT 5.01
INFO("recalc") might result in Automatic
INFO("release") might result in 11.0

INFO("system") might result in pcdos
INFO("totmem") might result in 2523040

end example]

18.17.7.169 INT

Syntax:

INT (*x*)

Description: Computes *x* rounded down to an integer.

Arguments:

Name	Type	Description
x	number	The value to be rounded down.

Return Type and Value: number – The rounded-down value of *x*.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`INT(8.9)` results in `8`
`INT(-8.9)` results in `-9`

end example]

18.17.7.170 INTERCEPT

Syntax:

`INTERCEPT` (*known-ys* , *known-xs*)

Description: Computes the point at which a line intersects the y-axis by using existing x-values and y-values. The intercept point is based on a best-fit regression line plotted through the known x-values and known y-values.

Mathematical Formula:

The equation for the intercept of the regression line, a, is:

$$a = \bar{y} - b\bar{x}$$

where the slope, b, is calculated as:

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*known-xs*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*known-ys*)

Arguments:

Name	Type	Description
known-ys	number, name, array, reference to number	The dependent set of observations or data. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
known-xs	number, name, array, reference to number	The independent set of observations or data. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The point at which a line intersects the y-axis by using existing x-values and y-values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

However, if

- *known-ys* and *known-xs* contain a different number of data points, the return value is unspecified.
- *known-ys* or *known-xs* contain no data points, the return value is unspecified.
- the line does not intersect the y-axis, #DIV/0! Is returned.

[*Example*:

`INTERCEPT({2,3,9,1,8},{6,5,11,7,5})` results in `0.048387097`

end example]

18.17.7.171 INTRATE

Syntax:

`INTRATE ( `*settlement*` , `*maturity*` , `*investment*` , `*redemption*` [ , [ `*basis*` ] ] )`

Description: Computes the interest rate for a fully invested security.

Mathematical Formula:

$$INTRATE = \frac{redemption - nvestment}{investment} \times \frac{B}{DIM}$$

where:

- B = number of days in a year, depending on the year basis
- DIM = number of days from settlement to maturity.
- *investment* = argument *investment*
- *redemption* = argument *redemption*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
investment	number	The amount invested in the security.
redemption	number	The amount to be received at maturity.
basis	number	The truncated integer type of day count basis to use, as follows: `0` or omitted: US (NASD) 30/360. Assumes that each

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description		
				month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
			1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
			2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			3	Actual/365. Similar to Basis 1, but always has 365 days per year.
			4	European 30/360. The European method for

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The interest rate for a fully invested security.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *investment* or *redemption* ≤ 0, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`INTRATE(DATE(2008,2,15),DATE(2008,5,15),1000000,1014420,2)` results in `5.7680%`

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.172 IPMT

Syntax:

```
IPMT ( rate , per, nper , pv , [ fv ] [ , [ type ]] )
```

Description: Computes the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate.

Arguments:

Name	Type	Description
rate	number	The interest rate.
per	number	The period for which the interest is to be found, and shall be in the range 1–*nper*.
nper	number	The total number of payment periods in an annuity.
pv	number	The present value, or the lump-sum amount that a series of future payments is worth right now.
fv	number	The future value, or a cash balance to be attained after the last payment is made. If omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0).
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
0	Payment at the end of the period
1	Payment at the beginning of the period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Arguments representing cash paid by investor shall be expressed as negative numbers; arguments representing cash received by the investor shall be expressed as positive numbers.

Return Type and Value: number – The interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate.

However, if *type* is any number other than 0 or 1, `#NUM!` is returned.

[*Example*:

`IPMT(0.1/12,1*3,3,8000)` results in `-22.41`

©ISO/IEC 2016 – All rights reserved

`IPMT(0.1,3,3,8000)` results in `-292.45`

end example]

18.17.7.173 IRR

`IRR ( values [ , [ guess ] ] )`

Description: Computes the internal rate of return for a series of cash flows represented by the numbers in *values*. (These cash flows do not have to be even, as they would be for an annuity. However, the cash flows shall occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods.)

Arguments:

Name	Type	Description
values	array, reference, text, logical	The set of numbers for which the internal rate of return is to be calculated. *values* shall contain at least one positive value and one negative value to calculate the internal rate of return. The order of numbers in *values* is significant, so be sure payment and income numbers are in the desired sequence. If *values* contains elements that are text, logical values, or empty cells, those elements are ignored.
guess	number	An estimate of the result of `IRR`. If omitted, it is assumed to be 0.1 (i.e., 10 percent).

Return Type and Value: number – The internal rate of return for a series of cash flows.

However, if the implementation determines that a return value cannot be computed, `#NUM!` is returned.

[*Example*:

`IRR({-70000,12000,15000,18000,21000})` results in `-2.1245%`
`IRR({-70000,12000,15000,18000,21000,26000})` results in `8.6631%`
`IRR({-70000,12000,15000},-0.1)` results in `-44.3507%`

end example]

18.17.7.174 ISBLANK

Syntax:

`ISBLANK ( value )`

Description: Determines if *value* refers to an empty cell.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* refers to an empty cell; otherwise, FALSE.

[*Example*:

`ISBLANK(A10)` results in TRUE, when A10 is empty
`ISBLANK(A10)` results in FALSE, when A10 contains 123

end example]

18.17.7.175 ISERR

Syntax:

`ISERR ( value )`

Description: Determines if *value* is any of the error values other than #N/A.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* is one of the error values, excluding #N/A; otherwise, FALSE.

[*Example*:

`ISERR(A1)` results in TRUE if A1 evaluates to `#DIV/0!`, for example
`ISERR(B1)` results in FALSE if B1 evaluates to `#N/A`

end example]

18.17.7.176 ISERROR

Syntax:

`ISERROR ( value )`

Description: Determines if *value* is any of the error values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* is one of the error values; otherwise, FALSE.

[*Example*:

`ISERROR(A1)` results in TRUE if A1 evaluates to `#DIV/0!`, for example

end example]

18.17.7.177 ISEVEN

Syntax:

```
ISEVEN ( value )
```

Description: Determines if *value* is an even number or refers to a cell containing an even number.

Arguments:

Name	Type	Description
value	number	The value to be tested. It is truncated to an integer.

Return Type and Value: logical – TRUE if *value* is an even number or refers to a cell containing an even number; otherwise, FALSE.

[*Example*:

`ISEVEN(12.456)` results in TRUE
`ISEVEN(A10)` results in FALSE, when A10 contains -15

end example]

18.17.7.178 ISLOGICAL

Syntax:

```
ISLOGICAL ( value )
```

Description: Determines if *value* contains a logical value or refers to a cell containing a logical value.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* contains a logical value or refers to a cell containing a logical value; otherwise, FALSE.

[*Example*:

ISLOGICAL(TRUE) results in TRUE
ISLOGICAL(A10) results in FALSE, when A10 contains 123
ISLOGICAL({TRUE,2}) results in TRUE
ISLOGICAL({2,TRUE}) results in FALSE

end example]

18.17.7.179 ISNA

Syntax:

```
ISNA ( value )
```

Description: Determines if *value* is the error value #N/A.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* is #N/A; otherwise, FALSE.

[*Example*:

ISERR(A1) results in TRUE if A1 evaluates to #N/A
ISERR(B1) results in TRUE if B1 evaluates to #DIV/0!, for example

end example]

18.17.7.180 ISNONTEXT

Syntax:

```
ISNONTEXT ( value )
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Determines if *value* does not contain text or does not refer to a cell containing text. An empty cell is not text.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* does not contain text or does not refer to a cell containing text; otherwise, FALSE.

[*Example*:

`ISNONTEXT("ABC")` results in FALSE
`ISNONTEXT(A10)` results in TRUE, when A10 contains 123
`ISNONTEXT({1,"ABC"})` results in TRUE
`ISNONTEXT({"ABC",1})` results in FALSE

end example]

18.17.7.181 ISNUMBER

Syntax:

```
ISNUMBER ( value )
```

Description: Determines if *value* contains a number or refers to a cell that contains a number.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* contains a number or refers to a cell that contains a number; otherwise, FALSE.

[*Example*:

`ISNUMBER(10.56)` results in TRUE
`ISNUMBER(A10)` results in FALSE, when A10 contains ABC
`ISNUMBER({1,"ABC"})` results in TRUE
`ISNUMBER({"ABC",1})` results in FALSE

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.182 ISO.CEILING

Syntax:

```
ISO.CEILING ( x , [ significance ] )
```

Description: Computes a value that is *x* rounded-up, to the nearest multiple of *significance*. Regardless of the sign of *x*, a value is rounded up.

Arguments:

Name	Type	Description
x	number	The value to be rounded
significance	number	The optional multiple to which *x* is to be rounded. If *significance* is omitted, its default value is 1. [*Note*: The absolute value of the multiple is used, so the CEILING function will return the mathematical ceiling irrespective of the signs of *x* and *significance*. *end note*]

Return Type and Value: number – The rounded-up value of *x*.

However, if *x* and/or *significance* is zero, zero is returned.

[*Example*:

`ISO.CEILING(4.3)` rounds 4.3 up to nearest multiple of 1; that is, to 5
`ISO.CEILING(-4.3)` rounds -4.3 up to nearest multiple of 1; that is, to -4
`ISO.CEILING(4.3, 2)` rounds 4.3 up to the nearest multiple of 2; that is, to 6
`ISO.CEILING(4.3,-2)` rounds 4.3 up to the nearest multiple of -2; that is, to 6
`ISO.CEILING(-4.3,2)` rounds -4.3 up to the nearest multiple of 2; that is, to -4
`ISO.CEILING(-4.3,-2)` rounds -4.3 up to the nearest multiple of -2; that is, to -4

end example]

18.17.7.183 ISODD

Syntax:

```
ISODD ( value )
```

Description: Determines if *value* is an odd number or refers to a cell containing an odd number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Arguments:

Name	Type	Description
value	number	The value to be tested. It is truncated to an integer.

Return Type and Value: logical – TRUE if *value* is an odd number or refers to a cell containing an odd number; otherwise, FALSE.

[*Example*:

`ISODD(12.456)` results in FALSE
`ISODD(A10)` results in TRUE, when A10 contains `-15`

end example]

18.17.7.184 ISPMT

Syntax:

`ISPMT ( rate , per, nper , pv )`

Description: Computes the interest paid during a specific period of an investment.

Arguments:

Name	Type	Description
rate	number	The interest rate for the investment.
per	number	The period for which the interest is to be found, and shall be in the range 1–*nper*.
nper	number	The total number of payment periods for the investment.
pv	number	The present value the investment.

Arguments representing cash paid by investor shall be expressed as negative numbers; arguments representing cash received by the investor shall be expressed as positive numbers.

Return Type and Value: number – The interest paid during a specific period of an investment.

[*Example*:

`ISPMT(0.1/12,1,3*12,8000000)` results in `-64814.81`
`ISPMT(0.1,1,3,8000000)` results in `-533333.33`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.185 ISREF

Syntax:

```
ISREF ( value )
```

Description: Determines if *value* is a cell reference.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* is a cell reference; otherwise, FALSE.

[*Example*:

`ISREF("ABC")` results in FALSE
`ISREF(A10)` results in TRUE

end example]

18.17.7.186 ISTEXT

Syntax:

```
ISTEXT ( value )
```

Description: Determines if *value* contains text or refers to a cell containing text.

Arguments:

Name	Type	Description
value	any	The value to be tested. No conversion shall take place on an argument passed to this function.

Return Type and Value: logical – TRUE if *value* contains text or refers to a cell containing text; otherwise, FALSE.

[*Example*:

`ISTEXT("ABC")` results in TRUE
`ISTEXT(A10)` results in FALSE, when A10 contains 123
`ISTEXT({1,"ABC"})` results in FALSE
`ISTEXT({"ABC",1})` results in TRUE

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.187 JIS

Syntax:

`JIS` (*string*)

Description: Creates a string that is the conversion of half-width (single-byte) letters within *string* to full-width (double-byte) characters.

Arguments:

Name	Type	Description
string	text	Designates the string to be converted. If *string* does not contain any half-width English letters or katakana, nothing in *string* is converted.

Return Type and Value: text – The string resulting from the conversion.

[*Example*:

`JIS("ABC")` results in ABC
`JIS("ｴｸｾﾙ")` results in エクセル

end example]

18.17.7.188 KURT

Syntax:

`KURT` (*argument-list*)

Description: Computes the kurtosis of a data set. Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution.

Mathematical Formula:

Kurtosis is defined as:

$$\left\{\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum\left(\frac{x_i-\bar{x}}{s}\right)^4\right\}-\frac{3(n-1)^2}{(n-2)(n-3)}$$

where:

n is the number of elements in *argument-list*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

xj is the value of the j-th element in *argument-list*.

$\bar{x}$ is the mean of the values in *argument-list*.

s is the standard deviation of the values in *argument-list*.

Arguments:

Name	Type	Description
argument-list	array reference to an array, number, name, or reference to number.	The *arguments* in *argument-list* are the values for which kurtosis is to be calculated. Any *argument* in *argument-list* can be an array or a reference to an array. Logical values and text representations of numbers that are directly entered into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The kurtosis of a data set.

However, if

- There are fewer than four data points, the return value is unspecified.
- The standard deviation of the sample equals zero, the return value is unspecified.

[*Example*:

```
KURT(10.5,12.4,19.4,23.2) results in  -3.644621343
KURT(10.5,{12.4,19.4},23.2) results in  -3.644621343
```

end example]

18.17.7.189 LARGE

Syntax:

```
LARGE ( array , k )
```

Description: Computes the k^{th} largest value in a data set.

Arguments:

Name	Type	Description
array	array, reference	The set of numbers from which the k^{th}-largest value is to be determined.
k	number	The position (from the largest) in the array or cell range of data to return.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The k^{th} largest value in a data set.

However, if

- *array* is empty, the return value is unspecified.
- $k \leq 0$, `#NUM!` is returned.
- k is greater than the number of data points, `#NUM!` is returned.

[*Example*:

`LARGE({3,5,3,5,4;4,2,4,6,7},3)` results in `5`
`LARGE({3,5,3,5,4;4,2,4,6,7},7)` results in `4`

end example]

18.17.7.190 LCM

Syntax:

`LCM` (*argument-list*)

Description: Computes the least common multiple of the one or more *argument*s in *argument-list*.

Arguments:

Name	Type	Description
argument-list	number	*argument-list* specifies the *argument*s. Each argument is truncated to an integer.

Return Type and Value: number – The least common multiple of one or more numbers.

However, if any *argument* is negative, `#NUM!` is returned.

[*Example*:

`LCM(5)` results in `5`
`LCM(5,2)` results in `10`
`LCM(24.99,36.45)` results in `72`
`LCM(24,36,15)` results in `360`

end example]

18.17.7.191 LEFT

Syntax:

`LEFT` (*string* [, *number-chars*])

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Extracts the left-most *number-chars* characters from *string*. (LEFT is intended for use with languages that use the single-byte character set (SBCS), whereas LEFTB (§18.17.7.192) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-chars	number	The number of characters to be extracted. If omitted, a count of 1 shall be assumed. *number-chars* shall be at least 0. If *number-chars* exceeds the length of *string*, the whole of *string* shall be extracted.

Return Type and Value: text – A string containing the left-most *number-chars* characters from *string*.

However, if *number-chars* is negative, #VALUE! is returned.

[*Example*:

`LEFT("abcdef",2)` results in `ab`
`LEFT(A10,4)` results in `xyz1`, when A10 contains `xyz123`

end example]

18.17.7.192 LEFTB

Syntax:

```
LEFTB ( string [ , number-bytes ] )
```

Description: Extracts the left-most *number-bytes*-worth of characters from *string*. (LEFTB is intended for use with languages that use the double-byte character set (DBCS), whereas LEFT (§18.17.7.192) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-bytes	number	The number of bytes to be extracted. If omitted, a count of 1 shall be assumed. *number-bytes* shall be at least 0. If *number-bytes* exceeds the length of *string*, the whole of *string* shall be extracted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: text – A string containing the left-most *number-bytes*-worth of characters from *string*.

However, if *number-bytes* is negative, #VALUE! is returned.

[*Example*: Assuming 1-byte characters:

`LEFTB("abcdef",2)` results in `ab`
`LEFTB(A10,4)` results in `xyz1`, when A10 contains `xyz123`

end example]

18.17.7.193 LEN

Syntax:

LEN (*string*)

Description: Determines the number of characters in *string*. (LEN is intended for use with languages that use the single-byte character set (SBCS), whereas LENB (§18.17.7.194) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string	text	Designates the string whose length is to be found.

Return Type and Value: number – The number of characters in *string*.

[*Example*:

`LEN("abc")` results in `3`
`LEN(A10)` results in `3`, when A1 contains `abc`

end example]

18.17.7.194 LENB

Syntax:

LENB (*string*)

Description: Determines the number of bytes in *string*. (LENB is intended for use with languages that use the double-byte character set (DBCS), whereas LEN (§18.17.7.193) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
string	text	Designates the string whose length is to be found.

Return Type and Value: number – The number of bytes in *string*.

[*Example*: Assuming 1-byte characters:

`LENB("abc")` results in 3
`LENB(A10)` results in 3, when A1 contains `abc`

end example]

18.17.7.195 LINEST

Syntax:

`LINEST ( known-ys [ , [ known-xs ] [ , [ const-flag ] [ , stats-flag ] ] )`

Description: Calculates the statistics for a line by using the "least squares" method to calculate a straight line that best fits the data, and returns an array that describes the line.

Mathematical Formula:

The equation for the line is:

$$y = mx + b$$

or

…

$y = m_1x_1 + m_2x_2 + \ldots + b$ (if there are multiple ranges of x-values)

where the dependent y-value is a function of the independent x-values. The m-values are coefficients corresponding to each x-value, and b is a constant value. y, x, and m can be vectors.

When there is only one independent x-variable, the slope and y-intercept values can be obtained directly by using the following formulas:

Slope: `INDEX(LINEST(`*known-ys*`,`*known-xs*`),1)`

Y-intercept: `INDEX(LINEST(`*known-ys*`,`*known-xs*`),2)`

The accuracy of the line calculated by `LINEST` depends on the degree of scatter in the data. The more linear the data, the more accurate the `LINEST` model. `LINEST` uses the method of least squares for determining the best

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

fit for the data. When there is only one independent x-variable, the calculations for m and b are based on the following formulas:

$$m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

$$b = \bar{y} - m\bar{x}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*known-xs*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*known-ys*)

Arguments:

Name	Type	Description
known-ys	array	The set of y-values already known in the relationship y=mx+b. If the array *known-ys* is a single column, then each column of *known-xs* is interpreted as a separate variable. If the array *known-ys* is a single row, then each row of *known-xs* is interpreted as a separate variable.
known-xs	array	An optional set of x-values that might already be known in the relationship y=mx+b. The array *known-xs* can include one or more sets of variables. If only one variable is used, *known-ys* and *known-xs* can be ranges of any shape, as long as they have equal dimensions. If more than one variable is used, *known-ys* shall be a vector (that is, a range with a height of one row or a width of one column). If *known-xs* is omitted, it is assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
const-flag	logical	Specifies whether to force the constant *b* to be zero. If TRUE or omitted, *b* is calculated normally. If FALSE, *b* is set to zero, and the m-values are adjusted to fit y=mx.
stats-flag	logical	Specifies whether to return additional regression statistics. If TRUE, LINEST returns the additional regression statistics (see table below), so the returned array is {mn, mn-1, …, m1, b; sen, sen-1, …, se1, seb; r2, sey; F, df; ssreg, ssresid}. If FALSE or omitted, LINEST returns only the m-coefficients and the constant b.

The additional regression statistics are as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Statistic	Description
se1, se2, …, se*n*	The standard error values for the coefficients m1, m2, …, m*n*.
seb	The standard error value for the constant b.
r2	The coefficient of determination.
sey	The standard error for the y estimate.
F	The F statistic, or the F-observed value.
df	The degrees of freedom.
ssreg	The regression sum of squares.
ssresid	The residual sum of squares.

Return Type and Value: array – The array that describes the line, in the form {mn, mn-1, …, m1, b}. The following illustration shows the order in which the additional regression statistics are returned.

[*Example*:

`LINEST({1,9,5,7},{0,4,2,3},,FALSE)` results in a slope of 2 and a y-intercept of 1

end example]

18.17.7.196 LN

Syntax:

`LN ( x )`

Description: Computes the natural logarithm of *x*.

Arguments:

Name	Type	Description
x	number	The positive real number for which the natural logarithm is being computed.

Return Type and Value: number – The natural logarithm of *x*.

However, if *x* is zero or negative, `#NUM!` is returned.

[*Example*:

`LN(86)` results in `4.454347296`
`LN(2.7182818)` results in `0.99999999`
`LN(EXP(3))` results in 3

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.197 LOG

Syntax:

```
LOG ( x [ , base ] )
```

Description: Computes the logarithm of *x* to the base *base*.

Arguments:

Name	Type	Description
x	number	The positive real number for which the logarithm is being computed.
base	number	The base of the logarithm. If omitted, base 10 is assumed.

Return Type and Value: number – The logarithm of *x*.

However, if

- *x* is zero or negative, `#NUM!` is returned.
- *base* is zero or negative, `#NUM!` is returned.
- *base* is `1`, `#DIV/0!` is returned.

[*Example*:

`LOG(10)` results in `1`
`LOG(8,2)` results in `3`
`LOG(86,2.7182818)` results in `4.454347343`

end example]

18.17.7.198 LOG10

Syntax:

```
LOG10 ( x )
```

Description: Computes the base-10 logarithm of *x*.

Arguments:

Name	Type	Description
x	number	The positive real number for which the logarithm is being computed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The base-10 logarithm of *x*.

However, if *x* is zero or negative, `#NUM!` is returned.

[*Example*:

`LOG10(86)` results in `1.934498451`
`LOG10(10)` results in `1`
`LOG10(1E5)` results in `5`
`LOG10(10^5)` results in `5`

end example]

18.17.7.199 LOGEST

Syntax:

`LOGEST (` *known-ys* `[ , [` *known-xs* `] [ , [` *const-flag* `] [ , [` *stats-flag* `] ] )`

Description: Calculates an exponential curve that fits the data, and returns an array of values that describes the curve.

Mathematical Formula:

The equation for the curve is:

$$y = b \times m^x$$

or

$$y = b \times {m_1}^{x_1} \times {m_2}^{x_2}$$ ... (if there are multiple x-values)

where the dependent y-value is a function of the independent x-values. The m-values are bases corresponding to each exponent x-value, and b is a constant value. [*Note*: y, x, and m can be vectors. *end note*]

When there is only one independent x-variable, the y-intercept (b) values can be obtained directly by using the following formula:

Y-intercept (b): `INDEX(LOGEST(`*known-ys*`,`*known-xs*`),2)`

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Name	Type	Description
known-ys	array	The set of y-values already known in the relationship $y=b*m^x$. If the array *known-ys* is a single column, then each column of *known-xs* is interpreted as a separate variable. If the array *known-ys* is a single row, then each row of *known-xs* is interpreted as a separate variable.
known-xs	array	An optional set of x-values that might already be known in the relationship $y=b*m^x$. The array *known-xs* can include one or more sets of variables. If only one variable is used, *known-ys* and *known-xs* can be ranges of any shape, as long as they have equal dimensions. If more than one variable is used, *known-ys* shall be a vector (that is, a range with a height of one row or a width of one column). If *known-xs* is omitted, it is assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
const-flag	logical	Specifies whether to force the constant *b* to be 1. If TRUE or omitted, *b* is calculated normally. If FALSE, *b* is set to 1, and the m-values are adjusted to fit $y=m^x$.
stats-flag	logical	Specifies whether to return additional regression statistics. If TRUE, LOGEST returns the additional regression statistics, so the returned array is {mn, mn-1, …, m1, b; sen, sen-1, …, se1, seb; r2, sey; F, df; ssreg, ssresid}. If FALSE or omitted, LOGEST returns only the m-coefficients and the constant b.

The additional regression statistics are described in §18.17.7.195.

Return Type and Value: array – The array that describes the line, in the form {mn, mn-1, …, m1, b}. The order in which the additional regression statistics are returned is described in §18.17.7.195.

[*Example*: Given the following data:

	A	B
1	Month	Units
2	11	33,100
3	12	47,300
4	13	69,000
5	14	102,000
6	15	150,000
7	16	220,000
8	Formula	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

	A	B
9	1.463275628	495.3047702

When `LOGEST(B2:B7,A2:A7,TRUE,FALSE)` is an array formula spanning cells A9:B9, those cells take on the results shown.

end example]

18.17.7.200 LOGINV

Syntax:

`LOGINV` (*probability* , *mean* , *standard-dev*)

Description: Calculates the inverse of the lognormal cumulative distribution function of x, where ln(x) is normally distributed with parameters *mean* and *standard-dev*.

Mathematical Formula:

$$LOGINV(p, \mu, \sigma) = e^{[\mu+\sigma\times(NORMSINV(p))]}$$

where:

- p = argument *probability*
- μ = argument *mean*
- σ = argument *standard-dev*

Arguments:

Name	Type	Description
probability	number	A probability associated with the lognormal distribution.
mean	number	The mean of ln(x).
standard-dev	number	The standard deviation of ln(x).

Return Type and Value: number – The inverse of the lognormal cumulative distribution function of x.

However, if

- *probability* ≤ 0 or *probability* ≥ 1, `#NUM!` is returned.
- *standard-dev* ≤ 0, `#NUM!` is returned.

[*Example*:

`LOGINV(0.039084,3.5,1.2)` results in `4.000025219`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

18.17.7.201 LOGNORMDIST

Syntax:

`LOGNORMDIST ( x , mean , standard-dev )`

Description: Calculates the cumulative lognormal distribution of x, where ln(x) is normally distributed with parameters *mean* and *standard-dev*.

Mathematical Formula:

$$LOGNORMDIST(x, \mu, \sigma) = NORMSDIST\left(\frac{\ln(x) - \mu}{\sigma}\right)$$

where:

- x = argument x
- μ = argument *mean*
- σ = argument *standard-dev*

Arguments:

Name	Type	Description
x	number	The value at which to evaluate the function.
mean	number	The mean of ln(x).
standard-dev	number	The standard deviation of ln(x).

Return Type and Value: number – The inverse of the lognormal cumulative distribution function of x.

However, if

- *x* ≤ 0, `#NUM!` is returned.
- *standard-dev* ≤ 0, `#NUM!` is returned.

[*Example*:

`LOGNORMDIST(4,3.5,1.2)` results in `0.039083556`

end example]

18.17.7.202 LOOKUP

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

vector form: LOOKUP (*lookup-value* , *lookup-vector* , *result-vector*)

array form: LOOKUP (*lookup-value* , *array*)

Description: The vector form looks in a vector for a value, and returns a value from the same position in a second vector. The array form looks in the first row or column of an array for the specified value and returns a value from the same position in the last row or column of that array.

Arguments:

Name	Type	Description
lookup-value	number, string, logical, name, reference	The value to search for in *lookup-vector* (or *array*).
lookup-vector	reference	A range that contains only one row or one column. The values in *lookup-vector* can be strings, numbers, or logical values. These values shall be placed in "ascending" order, as follows: …, -2, -1, 0, 1, 2, …, A–Z, FALSE, TRUE. Upper- and lowercase strings are equivalent. If LOOKUP can't find the *lookup-value*, it matches the largest value in *lookup-vector* (or *array*) that is less than or equal to *lookup-value*.
result-vector	reference	A range that contains only one row or column. It shall be the same size as *lookup-vector*.
array	text, number, logical	A range of cells whose values are to be compared with *lookup-value*. These values shall be placed in "ascending" order, as follows: …, -2, -1, 0, 1, 2, …, A–Z, FALSE, TRUE. Upper- and lowercase strings are equivalent. If *array* covers an area that has more columns than rows, *lookup-value* is searched for in the first row. If *array* is square or has more rows than columns, *lookup-value* is searched for in the first column.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: any – The vector form looks in a vector for a value, and returns a value from the same position in a second vector. The array form looks in the first row or column of an array for the specified value and returns a value from the same position in the last row or column of that array.

However, if

- *lookup-value* is smaller than the smallest value in *lookup-vector* (or the first row or column of *array*), the return value is unspecified.
- The size of the range specified by *result-vector* is not the same as that specified by *lookup-vector*, the return value is unspecified.
- The values in *lookup-vector* (or *array*) are not in "ascending" order, the return value is unspecified.

©ISO/IEC 2016 – All rights reserved

[*Example*: Given the following data:

	A	B
1	Frequency	Color
2	4.14	red
3	4.19	orange
4	5.17	yellow
5	5.77	green
6	6.39	blue

`LOOKUP(4.19,A2:A6,B2:B6)` results in `orange`
`LOOKUP(5,A2:A6,B2:B6)` results in `orange`
`LOOKUP(7.66,A2:A6,B2:B6)` results in `blue`

`LOOKUP("C",{"a","b","c","d";1,2,3,4})` results in 3
`LOOKUP("bump",{"a",1;"b",2;"c",3})` results in 2

end example]

18.17.7.203 LOWER

Syntax:

`LOWER ( string )`

Description: Makes a lowercase version of *string* by doing a character-by-character conversion of *string* to lowercase, except as noted below. [*Note*: The conversion of characters in *string* is not dependent on position/context of the character within the string, except as noted below. *end note*]

Arguments:

Name	Type	Description
string	text	Designates the string to be converted.

Return Type and Value: text – The lowercase version of *string* after doing a character-by-character conversion to lowercase. However, there is one exception; when Σ (U+03A3) is found in a word-final position, it is converted to ς (U+03C2) instead of σ (U+03C3).

[*Example*:

`LOWER("AbCd123#$%^")` results in `abcd123#$%^`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`LOWER(A10)` results in `234frtqwc$#%`, when A10 contains `234FRTqwc$#%`

end example]

18.17.7.204 MATCH

Syntax:

`MATCH ( lookup-value , lookup-array [ , [ match-type ] ] )`

Description: Locates the relative position of an array item that matches a specified value in a specified order. `MATCH` shall not distinguish between uppercase and lowercase letters when matching strings.

Arguments:

Name	Type	Description
lookup-value	number, string, logical, name, reference	The value to search for in *lookup-array*. If *match-type* is 0 and *lookup-value* is a string, the wildcard characters, question mark (?) and asterisk (*), can be used in *lookup-value*. A question mark matches any single character; an asterisk matches any sequence of characters. To locate a question mark or asterisk, precede that character with a tilde (~).
lookup-array	array, reference	A contiguous range of cells containing possible lookup values.
match-type	number	Specifies how *lookup-value* is matched with values in *lookup-array*, as follows:

Value	Meaning
-1	Finds the smallest value that is greater than or equal to *lookup-value*. The values in *lookup-array* shall be placed in "descending" order: TRUE, FALSE, Z–A, …, 2, 1, 0, -1, -2, …
0	Finds the first value that is exactly equal to *lookup-value*. The values in *lookup-array* can be in any order.
1 or omitted	Finds the largest value that is less than or equal to *lookup-value*. The values in *lookup-array* shall be placed in "ascending" order: …, -2, -1, 0, 1, 2, …, A–Z, FALSE, TRUE.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The relative position of an array item that matches a specified value in a specified order.

However, if

- No match is found, #N/A is returned.
- *match-type*'s value is out-of-bounds, `#NUM!` is returned.

[*Example*:

`MATCH(39,{25,38,40,41},1)` results in 2
`MATCH(41,{25,38,40,41},0)` results in 4

end example]

18.17.7.205 MAX

Syntax:

`MAX` (*argument-list*)

Description: Computes the largest of a set of numbers.

Arguments:

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number. Any *argument* can be an array or a reference to an array.	The *arguments* in *argument-list* designate the values for which the largest value is to be computed. Logical values and text representations of numbers occurring directly in the list of arguments are included. However, logical values and numbers in strings and are ignored inside references. [*Note*: To include these, use `MAXA` (§18.17.7.206). *end note*] If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The largest of a set of numbers; however, if the arguments contain no numbers, zero is returned.

[*Example*:

`MAX(10.4,-3.5,12.6)` results in `12.6`
`MAX(10.4,{-3.5,12.6})` results in `12.6`
`MAX({"ABC",TRUE})` results in `0`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Consider the case in which cell B3 contains 0:
MAX(-10,-12,-15,B3) results in -10
MAXA(-10,-12,-15,B3) results in 0

end example]

18.17.7.206 MAXA

Syntax:

MAXA (*argument-list*)

Description: Computes the largest of a set of numbers.

Arguments:

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number. Any *argument* can be an array or a reference to an array.	The *argument*s in *argument-list* designate the values for which the largest value is to be computed. Logical values and text representations of numbers occurring directly in the list of arguments are included. Logical values and numbers in strings inside references are also included. [*Note*: To ignore these, use MAX (§18.17.7.205). *end note*] If an array or reference argument contains non-numeric text or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The largest of a set of numbers; however, if the arguments contain no numbers, zero is returned.

[*Example*:

MAXA(10.4,-3.5,12.6) results in 12.6
MAXA(10.4,{-3.5,12.6}) results in 12.6
MAXA({"ABC",TRUE}) results in 0

Consider the case in which cell B3 contains 0:
MAX(-10,-12,-15,B3) results in -10
MAXA(-10,-12,-15,B3) results in 0

end example]

18.17.7.207 MDETERM

Syntax:

MDETERM (*array*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Computes the determinant of the square matrix of numbers designated by *array*. The determinant is calculated with an accuracy of at least 15 digits, which can lead to a small numeric error when the calculation is not complete. [*Example*: The determinant of a singular matrix can differ from zero by 1E-16. *end example*]

Arguments:

Name	Type	Description
array	array, reference	Designate a square matrix of numbers.

Return Type and Value: number – The determinant of *array*. Some square matrices cannot be inverted. The determinant of a non-invertible matrix is 0.

However, if

- Any cells in *array* are empty or contain text, the return value is unspecified.
- The matrix designated by *array* is not square, #VALUE! is returned.

[*Example*:

`MDETERM(A2:D5)` results in the determinant of the 4x4 array designated by the cell range
`MDETERM({3,6,1;1,1,0;3,10,2})` results in `1`
`MDETERM({3,6;1,1})` results in `-3`

end example]

18.17.7.208 MDURATION

Syntax:

`MDURATION ( `*settlement*` , `*maturity*` , `*coupon*` , `*yld*` , `*frequency*` [ , [ `*basis*` ] ] )`

Description: Computes the modified Macaulay duration for a security with an assumed par value of 100.

Mathematical Formula:

$$MDURATION = \frac{DURATION}{1 + \left(\frac{Market\ yield}{Coupon\ payments\ per\ yea}\right)}$$

where:

- *Coupon payments per year* = argument *frequency*
- *DURATION* = `DURATION(settlement, maturity, coupon, yld, frequency, [basis])`
- *Market yield* = argument *yld*

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>settlement</td><td>number</td><td>The security's settlement date.</td></tr>
<tr><td>maturity</td><td>number</td><td>The security's maturity date.</td></tr>
<tr><td>coupon</td><td>number</td><td>The security's annual coupon rate.</td></tr>
<tr><td>yld</td><td>number</td><td>The security's annual yield.</td></tr>
<tr><td>frequency</td><td>number</td><td>the number of coupon payments per year. (For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4.) frequency is truncated to an integer.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.</td></tr><tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Time information in the date arguments is ignored.

Return Type and Value: number – The modified Macaulay duration for a security with an assumed par value of 100.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *coupon* or *yld* < 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`MDURATION(DATE(2008,1,1),DATE(2016,1,1),0.08,0.09,2,1)` results in `5.7357`

end example]

18.17.7.209 MEDIAN

Syntax:

`MEDIAN ( argument-list )`

Description: Computes the median of the numeric values of its arguments. The median of a set of values is the value for which half the numbers in the set are greater, and half the values are less. For sets with an odd number of values, the median is calculated by finding the value whose rank in the ordered set of all values is equal to half the number of items (n) in the set plus one half (i.e., n/2 + 1/2). If the number of values in the set is even, then the median is defined to be the average of the values of rank n/2 and n/2 + 1.

Arguments:

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number.	The *argument*s in *argument-list* designate the values whose median is to be computed. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The median of the values of its arguments.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

`MEDIAN(10,20)` results in `15`
`MEDIAN(-3.5,1.4,6.9,-4.5)` results in `-1.05`
`MEDIAN({-3.5,1.4,6.9},-4.5)` results in `-1.05`

end example]

18.17.7.210 MID

Syntax:

`MID ( string , start-pos , number-chars )`

Description: Extracts *number-chars* characters from *string*, starting at character position *start-pos*. (`MID` is intended for use with languages that use the single-byte character set (SBCS), whereas `MIDB` (§18.17.7.211) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-chars	number	The number of characters to be extracted. *number-chars* shall be at least 0.
start-pos	number	The starting position within *string*, where the first character is position 1. If *start-pos* is greater than the length of *string*, or if *start-pos* and *number-chars* combined exceeds the length of *string*, the whole of *string* shall be extracted.

Return Type and Value: text – A string containing *number-chars* characters from *string*, starting at character position *start-pos*.

However, if

- *start-pos* < 1, #VALUE! is returned.
- *number-chars* < 0, #VALUE! is returned.

[*Example*:

`MID("abcdef",3,2)` results in `cd`
`MID(A10,4,1)` results in `1`, when A10 contains `xyz123`
`MID("abcdef",4,5)` results in `def`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.211 MIDB

Syntax:

```
MIDB ( string , start-pos , number-bytes )
```

Description: Extracts *number-bytes*-worth of characters from *string*, starting at character position *start-pos*. (MIDB is intended for use with languages that use the double-byte character set (DBCS), whereas MID (§18.17.7.210) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-bytes	number	The number of characters to be extracted. *number- bytes* shall be at least 0.
start-pos	number	The starting position within *string*, where the first byte is position 1. If *start-pos* is greater than the length of *string*, or if *start-pos* and *number- bytes* combined exceeds the length of *string*, the whole of *string* shall be extracted.

Return Type and Value: text – A string containing *number-bytes*-worth of characters from *string*, starting at character position *start-pos*.

However, if

- *start-pos* < 0, #VALUE! is returned.
- *number-bytes* < 0, #VALUE! is returned.

[*Example*: Assuming 1-byte characters:

`MIDB("abcdef",3,2)` results in `cd`
`MIDB(A10,4,1)` results in `1`, when A10 contains `xyz123`
`MIDB("abcdef",4,5)` results in `def`

end example]

18.17.7.212 MIN

Syntax:

```
MIN ( argument-list )
```

Description: Computes the smallest of a set of numbers.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number. Any *argument* can be an array or a reference to an array.	The *arguments* in *argument-list* designate the values for which the smallest value is to be computed. Logical values and text representations of numbers occurring directly in the list of arguments are included. However, logical values and numbers in strings and are ignored inside references. [*Note*: To include these, use MINA (§18.17.7.213). *end note*] If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The smallest of a set of numbers; however, if the *arguments* contain no numbers, zero is returned.

[*Example*:

`MIN(10.4,-3.5,12.6)` results in `-3.5`
`MIN(10.4,{-3.5,12.6})` results in `-3.5`
`MIN({"ABC",TRUE})` results in `0`

Consider the case in which cell `B3` contains `0`:
`MIN(10,12,15,B3)` results in `10`
`MINA(10,12,15,B3)` results in `0`

end example]

18.17.7.213 MINA

Syntax:

`MINA ( `*argument-list*` )`

Description: Computes the smallest of a set of numbers.

Arguments:

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number. Any *argument* can be an array or a reference to an array.	The *arguments* in *argument-list* designate the values for which the smallest value is to be computed. Logical values and text representations of numbers occurring directly in the list of arguments are included. Logical values and numbers in strings inside references are also included. [*Note*: To ignore these, use MIN (§18.17.7.212). *end note*] If an array or reference argument contains non-numeric text or empty cells, those values are ignored;

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		however, cells with the value 0 are included.

Any *argument* in *argument-list* can be an array or a reference to an array.

Return Type and Value: number – The smallest of a set of numbers; however, if the *argument*s contain no numbers, zero is returned.

[*Example*:

`MINA(10.4,-3.5,12.6)` results in `-3.5`
`MINA(10.4,{-3.5,12.6})` results in `-3.5`
`MINA({"ABC",TRUE})` results in `0`

Consider the case in which cell `B3` contains `0`:
`MIN(10,12,15,B3)` results in `10`
`MINA(10,12,15,B3)` results in `0`

end example]

18.17.7.214 MINUTE

Syntax:

`MINUTE (` *time-value* `)`

Description: Computes the minute for the date and/or time having the given *time-value*.

Arguments:

Name	Type	Description
time-value	number	The date and/or time whose minute is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its integer part is ignored, or as a *string-constant* having any date and/or time format, in which case, any date information shall be ignored.

Return Type and Value: number – The minute for the date and/or time having the given *time-value*. The returned value shall be in the range 0–59.

However, if *time-value* is out of range for the current date system, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

`MINUTE(DATE(2006,2,26)+TIME(2,10,20))` results in `10`
`MINUTE(TIME(22,56,34))` results in `56`
`MINUTE(0)` results in 0, since serial date-time 0 represents `00:00:00`
`MINUTE(10.5)` results in 0, since serial date-time .5 represents `12:00:00`
`MINUTE("22-Oct-2001 10:53:12")` results in `53`
`MINUTE("10:53:12 pm")` results in `53`
`MINUTE("22:53:12")` results in `53`

end example]

18.17.7.215 MINVERSE

Syntax:

`MINVERSE ( array )`

Description: Computes the inverse of the square matrix of numbers designated by *array*. The inverse matrix is calculated with an accuracy of at least 15 digits, which can lead to a small numeric error when the cancellation is not complete.

Arguments:

Name	Type	Description
array	array, reference	Designate a square matrix of numbers.

Return Type and Value: number – The inverse of the square matrix designated by *array*.

However, if

- Any cells in *array* are empty or contain text, the return value is unspecified.
- The matrix designated by *array* is not square, #VALUE! is returned.
- The matrix cannot be inverted, the return value is unspecified.

[*Example*:

`MINVERSE({3,6,1;1,1,0;3,10,2})` results in `2`
`MINVERSE({3,6;1,1})` results in `-0.333333333`

end example]

18.17.7.216 MIRR

`MIRR ( values , finance-rate , reinvest-rate )`

Description: Computes the modified internal rate of return for a series of periodic cash flows. (Both the cost of the investment and the interest received on reinvestment of cash are considered.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Mathematical Formula:

The formula for MIRR is:

$$\left(\frac{-NPV(rrate, values[positive]) * (1 + rrate)^{n}}{NPV(frate, values[negative]) * (1 + frate)}\right)^{\frac{1}{n-1}} - 1$$

where:

- *frate* = argument *finance-rate*
- *n* = number of cash flows in argument *values*
- *rrate* = argument *reinvest-rate*
- *values* = argument *values*

Arguments:

Name	Type	Description
values	array, reference	Designates a set of numbers for which the rate of return is to be calculated. *values* shall contain at least one positive value and one negative value to calculate the internal rate of return. The order of numbers in *values* is significant, so be sure payment and income numbers are in the desired sequence. If *values* contains elements that are text, logical values, or empty cells, those elements are ignored.
finance-rate	number	The interest rate paid pay on the money used in the cash flows.
reinvest-rate	number	The interest rate received on the cash flows as they are reinvested.

Return Type and Value: number – The modified internal rate of return for a series of periodic cash flows.

However, if *values* does not contain at least one positive value and one negative value, `#DIV/0!` is returned.

[*Example*:

`MIRR({-120000,39000,30000,21000,37000,46000},0.1,0.12)` results in `12.6094%`
`MIRR({-120000,39000,30000,21000},0.1,0.12)` results in `-4.8045%`
`MIRR({-120000,39000,30000,21000,37000,46000},0.1,0.14)` results in `13.4759%`

end example]

18.17.7.217 MMULT

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

MMULT (*array-1* , *array-2*)

Description: Computes the product of the matrices of numbers designated by *array-1* and *array-2*.

Mathematical Formula:

The matrix product array *a* of two arrays *b* and *c* is:

$$a_{ij} = \sum_{k=1}^{n} b_{ik}\, c_{kj}$$

Where:

- b_{ik} = the element in the i-th row and k-th column in argument *array-1*
- c_{kj} = the element in the k-th row and j-th column in argument *array-2*
- i = the row number
- j = the column number

Arguments:

Name	Type	Description
array-1	array, reference, name	Designate the matrices of numbers to be multiplied.
array-2		

Return Type and Value: number – The product of the matrices of numbers designated by *array-1* and *array-2*.

However, if

- Any cells in *array-1* or *array-2* are empty or contain text, the return value is unspecified.
- The number of columns in *array-1* is different from the number of rows in *array-2*, #VALUE! is returned.

[*Example*:

`MMULT({3,6,1;1,1,0},{5,7;4,6;2,5})` results in `41`

end example]

18.17.7.218 MOD

Syntax:

MOD (*x* , *y*)

Description: Computes the remainder when *x* is divided by *y*. The result has the same sign as *y*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
x	number	The number for which the remainder is being sought.
y	number	The number by which *x* is to be divided.

Return Type and Value: number – The remainder when *x* is divided by *y*. The result has the same sign as *y*. If *y* is 0, the return value is unspecified.

[*Example*:

`MOD(3,2)` results in `1`
`MOD(-3,2)` results in `1`
`MOD(3,-2)` results in `-1`
`MOD(-3,-2)` results in `-1`

end example]

18.17.7.219 MODE

Syntax:

`MODE` (*argument-list*)

Description: Computes the most frequently occurring of the numeric values of its arguments. If the set of values contains more than one most-frequent value, the first occurrence of any most-frequent value in the list is used as the result.

Arguments:

Name	Type	Description
argument-list	logical, number, name, arrays, reference to number. Any *argument* can be an array or a reference to an array.	The *argument*s in *argument-list* designate the values whose mode is to be computed. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The most frequently occurring of the values of its arguments.

However, if the data set contains no duplicate data points, `#N/A` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

MODE(9,1,5,1,9,5,6,6) results in 9
MODE(1,9,5,1,9,5,6,6) results in 1
MODE(5,1,9,5,1,9,6,6) results in 5

end example]

18.17.7.220 MONTH

Syntax:

MONTH (*date-value*)

Description: Computes the numeric month in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*, taking into account the current date system. That date and/or time shall be expressed either as a serial date-time, in which case, its fractional part is ignored, or as a *string-constant* having any date and/or time format, in which case, any time information shall be ignored.

Arguments:

Name	Type	Description
date-value	number, text	The date and/or time whose month is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its fractional part is ignored, or as a *string-constant* having any date and/or time format, in which case, any time information shall be ignored.

Return Type and Value: number – The month in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*, in the range 1–12.

However, if *date-value* is out of range for the current date system, #NUM! is returned.

[*Example*:

MONTH(DATE(2006,1,2)) results in 1
MONTH(DATE(2006,0,2)) results in 12
MONTH("2006/1/2 10:45 AM") results in 1
MONTH(30000) results in 2 for both the 1900 and 1904 date systems

end example]

18.17.7.221 MROUND

Syntax:

MROUND (*x* , *multiple*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Computes *x* rounded to *multiple*, away from zero. It rounds up if the remainder of dividing *x* by *multiple* is greater than or equal to half the value of *multiple*; otherwise, it rounds down.

Arguments:

Name	Type	Description
x	number	The value to round.
multiple	number	The multiple to which *x* is to be rounded.

Return Type and Value: number – *x* rounded to *multiple*.

However, if *x* and *multiple* have different signs, `#NUM!` is returned.

[*Example*:

`MROUND(10,3)` rounds 10 to a nearest multiple of 3; that is, to 9
`MROUND(-10,-3)` rounds -10 to a nearest multiple of -3; that is, to `-9`
`MROUND(1.3,0.2)` rounds 1.3 to a nearest multiple of 0.2; that is, to `1.4`

end example]

18.17.7.222 MULTINOMIAL

Syntax:

`MULTINOMIAL (` *argument-list* `)`

Description: Computes the ratio of the factorial of the sum of the values in *argument-list* to the product of the factorials.

Mathematical Formula:

The multinomial is:

$$MULTINOMIAL(a, b, c) = \frac{(a + b + c)!}{a!\, b!\, c!}$$

where:

- a, b, c, … = the elements in *argument-list*

Arguments:

Name	Type	Description
argument-list	number	The *arguments* in *argument-list* designate the numerical values for which the multinomial is desired.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The ratio of the factorial of the sum of the values in *argument-list* to the product of the factorials.

However, if any *argument* is less than zero, `#NUM!` is returned.

[*Example*:

`MULTINOMIAL(2)` results in `1`
`MULTINOMIAL(2,3)` results in `10`
`MULTINOMIAL(2,3,4)` results in `1260`

end example]

18.17.7.223 N

Syntax:

`N ( value )`

Description: Converts *value* to a number or, if *value* is a reference to a single cell, converts the value of that cell to a number.

Arguments:

Name	Type	Description
value	any	Value to be converted.

Return Type and Value: number or error – An integer that is the converted value of *value*, or, if *value* is a reference to a single cell, the converted value of that cell, as follows:

value	Value Returned
number	That number
TRUE	1
FALSE	0
error value	That error value
Anything else (including array and text)	0

[*Example*:

`N(10.5)` results in `10.5`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`N(A10)` results in `-1234`, when A10 contains the number `-1234`
`N("ABC")` results in `0`
`N(A10)` results in `0`, when A10 contains the string ABC
`N(TRUE)` results in `1`
`N(A10)` results in `0`, when A10 contains `FALSE`
`N(A10)` results in `#N/A`, when A10 contains `#N/A`
`N({12.5,13.6,56.9})` results in `12.50`
`N(A10:A11)` results in `0`, when A10 contains `FALSE`, and A11 contains 321

end example]

18.17.7.224 NA

Syntax:

```
NA ( )
```

Description: Gets the error value `#N/A`. (The error value `#N/A` can be used instead of a call to this function; the result is the same.)

Arguments: None.

Return Type and Value: error – The error value `#N/A`.

[*Example*:

`NA()` results in `#N/A`
`IF(ISNA(NA()),"T","F")` results in `T`

end example]

18.17.7.225 NEGBINOMDIST

Syntax:

`NEGBINOMDIST` (*number-failures* , *number-successes* , *success-probability*)

Description: Computes the negative binomial distribution. `NEGBINOMDIST` returns the probability that there are *number-failures* failures before the *number-successes*th success, when the constant probability of a success is *success-probability*.

Mathematical Formula:

$$nb(x,r,p) = \binom{x+r-1}{r-1} p^r (1-p)^x$$

where:

- p = the argument *success-probability*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- *r* = the argument *number-successes*
- *x* = the argument *number-failures*

Arguments:

Name	Type	Description
number-failures	number	The number of failures, truncated to integer.
number-successes	number	The threshold number of successes, truncated to integer.
success-probability	number	The probability of a success.

Return Type and Value: number – The negative binomial distribution.

However, if

- *number-failures* < 0 or *number-successes* < 1, `#NUM!` is returned.
- *success-probability* ≤ 0 or *success-probability* ≥ 1, `#NUM!` is returned.

[*Example*:

`NEGBINOMDIST(6,10,0.5)` results in `0.076370239`

end example]

18.17.7.226 NETWORKDAYS

Syntax:

```
NETWORKDAYS ( start-date , end-date [ , holidays ] )
```

Description: Computes the number of whole working days between *start-date* and *end-date*. Weekend days (Saturday, Sunday) and any holidays specified by *holidays* are not considered as working days.

Arguments:

Name	Type	Description
start-date	number	The dates for which the difference is to be computed. *start-date* can be earlier than, the same as, or later than *end-date*.
end-date	number	
holidays	reference, array	An optional set of one or more dates that are to be excluded from the working day calendar. *holidays* shall be a range of cells that contain the dates, or an array constant of the serial date-times that represent those

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		dates. The ordering of dates or serial date-times in *holidays* can be arbitrary.

Return Type and Value: number – The number of whole working days between *start-date* and *end-date*, excluding the specified holidays. If *start-date* is later than *end-date*, the return value shall be negative, and the magnitude shall be the number of whole working days.

However, if

- *start-date* is out of range for the current date system, `#NUM!` is returned.
- *end-date* is out of range for the current date system, `#NUM!` is returned.

[*Example*:

`NETWORKDAYS(DATE(2006,1,1),DATE(2006,1,31))` results in 23
`NETWORKDAYS(DATE(2006,1,31),DATE(2006,1,1))` results in -23
`NETWORKDAYS(DATE(2006,1,1),DATE(2006,2,1),{"2006/1/2","2006/1/16"})` results in 21

end example]

18.17.7.227 NETWORKDAYS.INTL

Syntax:

Number form: `NETWORKDAYS.INTL (` *start-date* `,` *end-date* `[ , [` *weekend-number* `][ , ` *holidays* `]] )`

String form: `NETWORKDAYS.INTL (` *start-date* `,` *end-date* `[ , [` *weekend-string* `][ , ` *holidays* `]] )`

Description: Computes the number of whole working days between *start-date* and *end-date*. Weekend days and any holidays specified by *holidays* are not considered as working days.

Arguments:

Name	Type	Description
start-date	Number	The dates for which the difference is to be computed. *start-date* can be earlier than, the same as, or later than *end-date*.
end-date	Number	
weekend-number	Number	Indicates the days of the week that are weekend days and are not included in the number of whole working days between *start-date* and *end-date*. Values are shown in the table below.
weekend-string	String	Indicates the days of the week that are weekend days and are not included in the number of whole working days between *start-date* and *end-date*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		Values of *weekend-string* are seven characters long and each character in the string represents a day of the week, beginning with Monday. [*Example*: "0000011" would result in a weekend that is Saturday and Sunday. *end example*]
holidays	reference, array	An optional set of one or more dates that are to be excluded from the working day calendar. *holidays* shall be a range of cells that contain the dates, or an array constant of the serial date-times that represent those dates. The ordering of dates or serial date-times in *holidays* can be arbitrary.

weekend-number	Weekend days
1 or omitted	Saturday, Sunday
2	Sunday, Monday
3	Monday, Tuesday
4	Tuesday, Wednesday
5	Wednesday, Thursday
6	Thursday, Friday
7	Friday, Saturday
11	Sunday only
12	Monday only
13	Tuesday only
14	Wednesday only
15	Thursday only
16	Friday only
17	Saturday only

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The number of whole working days between *start-date* and *end-date*, excluding the specified weekend days and holidays. If *start-date* is later than *end-date*, the return value shall be negative, and the magnitude shall be the number of whole working days.

However, if

- *start-date* is out of range for the current date system, `#NUM!` is returned.
- *end-date* is out of range for the current date system, `#NUM!` is returned.

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`NETWORKDAYS.INTL(DATE(2006,1,1),DATE(2006,1,31))` results in 23
`NETWORKDAYS.INTL(DATE(2006,1,31),DATE(2006,1,1))` results in -23
`NETWORKDAYS.INTL(DATE(2006,1,1),DATE(2006,2,1),7,{"2006/1/2","2006/1/16"})` results in 21
`NETWORKDAYS.INTL(DATE(2006,1,1),DATE(2006,2,1),"0000110",{"2006/1/2","2006/1/16"})` results in 21

end example]

18.17.7.228 NOMINAL

`NOMINAL` (*effect-rate* , *npery*)

Description: Computes the nominal annual interest rate, given the effective rate and the number of compounding periods per year.

Mathematical Formula:

NOMINAL is related to EFFECT:

$$EFFECT = \left(1 + \frac{Nominal_rate}{Npery}\right)^{Npery} - 1$$

where:

- *Nominal_rate* = argument *effect-rate*
- *Npery* = argument *npery*

Arguments:

Name	Type	Description
effect-rate	number	The effective interest rate.
npery	number	The number of compounding periods per year, truncated to integer.

Return Type and Value: number – The nominal annual interest rate.

However, if

- *effect-rate* ≤ 0, `#NUM!` is returned.
- *npery* < 1, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

`NOMINAL(0.053543,4)` results in `5.2500%`

end example]

18.17.7.229 NORMDIST

Syntax:

`NORMDIST ( x , mean , standard-deviation , cumulative-flag )`

Description: Computes the normal distribution for the specified mean and standard deviation.

Mathematical Formula:

The equation for the normal density function (*cumulative-flag* = FALSE) is:

$$f(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

When *cumulative-flag* = TRUE, the formula is the integral from negative infinity to x of the given formula.

where:

- x = argument *x*
- μ = argument *mean*
- σ = argument *standard-dev*

Arguments:

Name	Type	Description
x	number	The value for which the distribution is to be computed.
mean	number	The arithmetic mean of the distribution.
standard-deviation	number	The standard deviation of the distribution.
cumulative-flag	logical	Determines the form of the function. If TRUE, then the cumulative distribution function is returned; if FALSE, the probability mass function is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The normal distribution for the specified mean and standard deviation.

However, if *standard-deviation* ≤ 0, `#NUM!` is returned.

[*Example*:

`NORMDIST(42,40,1.5,TRUE)` results in `0.90878878`
`NORMDIST(42,40,1.5,FALSE)` results in `0.10934005`

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.230 NORMINV

Syntax:

NORMINV (*probability* , *mean* , *standard-deviation*)

Description: Computes the inverse of the normal distribution for the specified mean and standard deviation.**Arguments:**

Name	Type	Description
probability	number	The probability corresponding to the normal distribution.
mean	number	The arithmetic mean of the distribution.
standard-deviation	number	The standard deviation of the distribution.

Return Type and Value: number – The inverse of the normal distribution for the specified mean and standard deviation.

However, if

- *probability* ≤ 0 or if *probability* ≥ 1, #NUM! is returned.
- *standard-deviation* ≤ 0, #NUM! is returned.
- the implementation determines that a return value cannot be computed, #N/A is returned.

[*Example*:

NORMINV(0.908789,40,1.5) results in 42.00000201

end example]

18.17.7.231 NORMSDIST

Syntax:

NORMSDIST (*z*)

Description: Computes the standard normal distribution for the specified mean and standard deviation.

Mathematical Formula:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

where:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- *z* = argument *z*

Arguments:

Name	Type	Description
z	number	The value for which the distribution is to be computed.

Return Type and Value: number – The standard normal distribution for the specified mean and standard deviation.

[*Example*:

`NORMSDIST(1.333333)` results in `0.90878873`
`NORMSDIST(-1.5)` results in `0.06680720`

end example]

18.17.7.232 NORMSINV

Syntax:

`NORMSINV ( probability )`

Description: Computes the inverse of the standard normal distribution. The distribution has a mean of zero and a standard deviation of 1.

Arguments:

Name	Type	Description
probability	number	The probability corresponding to the normal distribution.

Return Type and Value: number – The inverse of the standard normal distribution.

However, if

- *probability* ≤ 0 or if *probability* ≥ 1, `#NUM!` is returned.
- the implementation determines that a return value cannot be computed, `#N/A` is returned.

[*Example*:

`NORMSINV(0.945)` results in `1.59819314`
`NORMSINV(0.13)` results in `-1.12639113`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.233 NOT

Syntax:

NOT (*logical-value*)

Description: Computes the logical negation of *logical-value*.

Arguments:

Name	Type	Description
logical-value	logical	The value to be negated.

Return Type and Value: logical – The logical negation of *logical-value*; that is, it returns TRUE if *logical-value* is FALSE, and FALSE if *logical-value* is TRUE.

[*Example*:

NOT(TRUE) results in FALSE
NOT(FALSE) results in TRUE
NOT(10>5) results in FALSE
NOT(16.567) results in FALSE

end example]

18.17.7.234 NOW

Syntax:

NOW ( )

Description: Computes the serial date-time of the current date and time, taking into account the current date system.

Arguments: None.

Return Type and Value: number – The serial date-time of the current date and time.

[*Example*: On February 26, 2006, between 23:01 and 23:02, NOW() resulted in 38774.95958611110 for the 1900 date system. On February 26, 2006, between 23:02 and 23:03, NOW() resulted in 37312.95982569440 for the 1904 date system. *end example*]

18.17.7.235 NPER

Syntax:

NPER (*rate* , *pmt* , *pv* [, [*fv*] [, [*type*]]])

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Computes the number of periods for an investment based on periodic, constant payments and a constant interest rate.

Arguments:

Name	Type	Description
rate	number	The interest rate per period.
pmt	number	The payment made each period; it cannot change over the life of the annuity. Typically, *pmt* contains principal and interest but no other fees or taxes.
pv	number	The present value, or the lump-sum amount that a series of future payments is worth right now.
fv	number	The future value, or a cash balance to be attained after the last payment is made. If *fv* is omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0).
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
`0`	Payment at the end of the period
`1`	Payment at the beginning of the period

Return Type and Value: number – The number of periods for an investment based on periodic, constant payments and a constant interest rate.

However, if *type* is any number other than 0 or 1, `#NUM!` is returned.

[*Example*:

`NPER(0.12/12,-100,-1000,10000,1)` results in `59.67`
`NPER(0.12/12,-100,-1000)` results in `-9.58`

end example]

18.17.7.236 NPV

Syntax:

NPV (*rate* , *argument-list*)

Description: Calculates the net present value of an investment by using a discount rate and a series of future payments and income.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The NPV investment begins one period before the date of the first *argument* cash flow and ends with the last cash flow in the list. The calculation is based on future cash flows. If the first cash flow occurs at the beginning of the first period, the first value shall be added to the NPV result, not included in *argument-list*.

Mathematical Formula:

If n is the number of cash flows in the list of values:

$$NPV = \sum_{i=1}^{n} \frac{values_i}{(1 + rate)^i}$$

where:

n is the number of elements in *argument-list*.
values$_i$valuesj is the value of the i-th element in *argument-list*.
rate is the *rate* argument.

Arguments:

Name	Type	Description
rate	number	The rate of discount over the length of one period.
argument-list	number	The *arguments* in *argument-list* designate the series of future payments (negative values) and income (positive values). *arguments* shall be equally spaced in time and occur at the end of each period. The order of *arguments* is significant. *arguments* that are numbers, empty cells, logical values, or text representations of numbers are included; *arguments* that are error values or text that cannot be translated into numbers are ignored. If an *argument* is an array or reference, only numbers in that array or reference are included. Empty cells, logical values, text, or error values in the array or reference are ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – Net present value of an investment by using a discount rate and a series of future payments and income.

[*Example*:

`NPV(0.1,-10000,3000,4200,6800)` results in `1188.44`

end example]

18.17.7.237 OCT2BIN

Syntax:

OCT2BIN (*number* [, *num-bin-digits*])

Description: Makes the binary equivalent of *number*, with the result having *num-bin-digits* digits.

Arguments:

Name	Type	Description
number	text	A 10-digit octal number in a string that is to be converted to a binary string. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (30th bit from the right) representing the sign bit.
num-bin - digits	number	The number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-bin-digits* is ignored and the result has 10 digits. If *num-bin-digits* is omitted, the minimum number of digits is used in the result. *num-bin-digits* is truncated to an integer.

Return Type and Value: text – The binary equivalent of *number*.

However, if

- *number* is outside the range "7777777000" (11111111111111111111000000000 binary, -512 decimal) to "777" (00000000000000000000111111111 binary, 511 decimal), inclusive, #NUM! is returned.
- *number* contains one or more non-octal digits, #NUM! is returned.
- *number* contains more than 10 octal digits, #NUM! is returned.
- *number* needs more digits that *num-bin-digits*, #NUM! is returned.
- *num-bin-digits* ≤ 0 or > 10, #NUM! is returned.

[*Example*:

OCT2BIN("67") results in 110111
OCT2BIN("7777777776") results in 1111111110
OCT2BIN("7",5) results in 00111

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.238 OCT2DEC

Syntax:

OCT2DEC (*number*)

Description: Makes the decimal equivalent of *number*.

Arguments:

Name	Type	Description
number	text	A 10-digit octal number in a string that is to be converted to a decimal number. If *number* has less than 10 digits, leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (30th bit from the right) representing the sign bit.

Return Type and Value: number – The decimal equivalent of *number*.

However, if

- *number* contains one or more non-octal digits, #NUM! is returned.
- *number* contains more than 10 octal digits; that is, *number* is outside the range "4000000000" (-536,870,912 decimal) to "3777777777" (536,870,911 decimal), inclusive, #NUM! is returned.

[*Example*:

OCT2DEC("67") results in 55
OCT2DEC("7777777776") results in -2
OCT2DEC("7000000000") results in -134217728

end example]

18.17.7.239 OCT2HEX

Syntax:

OCT2HEX (*number* [, *num-hex-digits*])

Description: Makes the hexadecimal equivalent of *number*, with the result having *num-hex-digits* digits.

Arguments:

Name	Type	Description
number	text	A 10-digit octal number in a string that is to be converted to a hexadecimal string. If *number* has less than 10 digits,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		leading zero digits are implied until it has exactly 10 digits. The 10 digits use twos-complement representation with the left-most bit (30th bit from the right) representing the sign bit.
num-hex-digits	number	*num-hex-digits* is the number of digits in the result, with leading zeros added as necessary. However, if *number* is negative, *num-hex-digits* is ignored and the result has 10 digits. If *num-hex-digits* is omitted, the minimum number of digits is used in the result. *num-hex-digits* is truncated to an integer.

Return Type and Value: text – The hexadecimal equivalent of *number*.

However, if

- *number* contains one or more non-octal digits, `#NUM!` is returned.
- *number* contains more than 10 octal digits; that is, *number* is outside the range `"4000000000"` (20000000 hex, -536,870,912 decimal) to `"3777777777"` (1FFFFFFF hex, 536,870,911 decimal), inclusive, `#NUM!` is returned.
- *number* needs more digits that *num-hex-digits*, `#NUM!` is returned.
- *num-hex-digits* ≤ 0or > 10, `#NUM!` is returned.

[*Example*:

```
OCT2HEX("777") results in  1FF
OCT2HEX("7777777776") results in  FFFFFFFFFE
OCT2HEX("7",5) results in  00007
```

end example]

18.17.7.240 ODD

Syntax:

ODD (*x*)

Description: Computes *x* rounded to the nearest odd integer, away from zero. Regardless of the sign of *x*, a value is rounded up when adjusted away from zero.

Arguments:

Name	Type	Description
x	number	The value to be rounded.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The rounded value of *x*.

[*Example*:

`ODD(1.5)` rounds 1.5 up to the nearest odd integer; that is, to 3
`ODD(3)` rounds 3 up to the nearest odd integer; that is, to 3
`ODD(2)` rounds 2 up to the nearest odd integer; that is, to 3
`ODD(-1)` rounds -1 up to the nearest odd integer; that is, to `-1`
`ODD(-2)` rounds -2 up to the nearest odd integer; that is, to `-3`

end example]

18.17.7.241 ODDFPRICE

Syntax:

`ODDFPRICE` (*settlement* , *maturity* , *issue* , *first-coupon* , *rate* , *yld* , *redemption* , *frequency* [, [*basis*]])

Description: Computes the price per 100 currency units face value of a security having an odd (short or long) first period.

Mathematical Formula:

Odd short first coupon:

$$ODDFPRICE = \left[\frac{redemption}{\left(1+\frac{yld}{frequency}\right)^{\left(N-1+\frac{DSC}{E}\right)}}\right] + \left[\frac{100 \times \frac{rate}{frequency} \times \frac{DFC}{E}}{\left(1+\frac{yld}{frequency}\right)^{\frac{DSC}{E}}}\right] + \left[\sum_{k=2} \frac{100 \times \frac{rate}{frequency}}{\left(1+\frac{yld}{frequency}\right)^{\left(k-1+\frac{DSC}{E}\right)}}\right] - \left[100 \times \frac{rate}{frequency} \times \frac{A}{E}\right]$$

where:

- A = number of days from the beginning of the coupon period to the settlement date (accrued days).
- DFC = number of days from the beginning of the odd first coupon to the first coupon date.
- DSC = number of days from the settlement to the next coupon date.
- E = number of days in the coupon period.
- *frequency* = argument *frequency*
- N = number of coupons payable between the settlement date and the redemption date. (If this number contains a fraction, it is raised to the next whole number.)
- *rate* = argument *rate*
- *redemption* = argument *redemption*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- *yld* = argument *yld*

Odd long first coupon:

$$\text{ODDFPRICE} = \left[\frac{\text{redemption}}{\left(1+\frac{\text{yld}}{\text{frequency}}\right)^{\left(N+N_q+\frac{DSC}{E}\right)}}\right] + \left[\frac{100\times\frac{\text{rate}}{\text{frequency}}\times\left[\sum_{i=1}^{NC}\frac{DC_i}{NL_i}\right]}{\left(1+\frac{\text{yld}}{\text{frequency}}\right)^{N_q+\frac{DSC}{E}}}\right] + \left[\sum_{k=1}^{N}\frac{100\times\frac{\text{rate}}{\text{frequency}}}{\left(1+\frac{\text{yld}}{\text{frequency}}\right)^{\left(k-N_q+\frac{DSC}{E}\right)}}\right] - \left[100\times\frac{\text{rate}}{\text{frequency}}\times\sum_{i=1}^{NC}\frac{A_i}{NL_i}\right]$$

where:

- A_i = number of days from the beginning of the ith, or last, quasi-coupon period within odd period.
- DC_i = number of days from dated date (or issue date) to first quasi-coupon (i = 1) or number of days in quasi-coupon (i = 2,…, i = NC).
- *DSC* = number of days from settlement to next coupon date.
- *E* = number of days in coupon period.
- *frequency* = argument *frequency*
- *N* = number of coupons payable between the first real coupon date and redemption date. (If this number contains a fraction, it is raised to the next whole number.)
- *NC* = number of quasi-coupon periods that fit in odd period. (If this number contains a fraction, it is raised to the next whole number.)
- NL_i = normal length in days of the full ith, or last, quasi-coupon period within odd period.
- N_q = number of whole quasi-coupon periods between settlement date and first coupon.
- *rate* = argument *rate*
- *redemption* = argument *redemption*
- *yld* = argument *yld*

…**Arguments**:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
issue	number	The security's issue date.
first-coupon	number	The security's first coupon date.
rate	number	The security's interest rate.
yld	number	The security's annual yield.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>redemption</td><td>number</td><td>The security's redemption value per 100 currency units face value.</td></tr>
<tr><td>frequency</td><td>number</td><td>the number of coupon payments per year. (For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4.)</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>year is 366 days; otherwise it is 365 days.</td></tr><tr><td>2</td><td>Actual/360. Similar to Basis 1, but only has 360 days per year.</td></tr><tr><td>3</td><td>Actual/365. Similar to Basis 1, but always has 365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The price per 100 currency units face value of a security having an odd (short or long) first period.

 ©ISO/IEC 2016 – All rights reserved

However, if

- *settlement*, *maturity*, *issue*, or *first-coupon* is out of range for the current date system, `#NUM!` is returned.
- The following is not true: *maturity* is later than *first-coupon*, which is later than *settlement*, which is later than *issue*, so `#NUM!` is returned.
- *rate* or *yld* < 0, `#NUM!` is returned.
- *redemption* ≤ 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`ODDFPRICE(DATE(2008,11,11),DATE(2021,3,1),DATE(2008,10,15),DATE(2009,3,1), 0.0785,0.0625,100,2,1)` results in `113.5977`

end example]

18.17.7.242 ODDFYIELD

Syntax:

`ODDFYIELD ( `*settlement*` , `*maturity*` , `*issue*` , `*first-coupon*` , `*rate*` , `*pr*` , `*redemption*` , `*frequency*` [ , [ `*basis*` ] ] )`

Description: Computes the yield of a security that has an odd (short or long) first period.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
issue	number	The security's issue date.
first-coupon	number	The security's first coupon date.
rate	number	The security's interest rate.
pr	number	The security's price.
redemption	number	The security's redemption value per 100 currency units face value.
frequency	number	the number of coupon payments per year. (For annual payments, *frequency* is 1; for semiannual payments, *frequency* is 2; for quarterly payments, *frequency* is 4.)
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Name	Type	Description	
		0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The yield of a security that has an odd (short or long) first period.

However, if

- *settlement*, *maturity*, *issue*, or *first-coupon* is out of range for the current date system, `#NUM!` is returned.
- The following is not true: *maturity* is later than *first-coupon*, which is later than *settlement*, which is later than *issue*, so `#NUM!` is returned.
- *rate* or *pr* < 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

©ISO/IEC 2016 – All rights reserved

[*Example*:

```
ODDFYIELD(DATE(2008,11,11),DATE(2021,3,1),DATE(2008,10,15),DATE(2009,3,1),
  0.0575,84.5,100,2,0) results in 7.7246%
```

end example]

18.17.7.243 ODDLPRICE

Syntax:

```
ODDLPRICE ( settlement , maturity , last-interest , rate , yld , redemption ,
    frequency [ , [ basis ] ] )
```

Description: Computes the price per 100 currency units face value of a security having an odd (short or long) last coupon period.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
last-interest	number	The security's last coupon date.
rate	number	The security's interest rate.
yld	number	The security's annual yield.
redemption	number	The security's redemption value per 100 currency units face value.
frequency	number	the number of coupon payments per year. (For annual payments, *frequency* is 1; for semiannual payments, *frequency* is 2; for quarterly payments, *frequency* is 4.)
basis	number	The truncated integer type of day count basis to use, as follows: **Value** / **Day Count Basis** `0` or omitted / US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description		
				30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
			1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
			2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			3	Actual/365. Similar to Basis 1, but always has 365 days per year.
			4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

Return Type and Value: number – The price per 100 currency units face value of a security having an odd (short or long) last coupon period.

However, if

- *settlement*, *maturity*, or *last-interest* is out of range for the current date system, `#NUM!` is returned.
- The following is not true: *maturity* is later than *settlement*, which is later than *last-interest*, so `#NUM!` is returned.
- *rate* or *yld* < 0, `#NUM!` is returned.
- *redemption* ≤ 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

```
ODDLPRICE(DATE(2008,11,11),DATE(2021,3,1),DATE(2008,10,15),
  0.0785,0.0625,100,2,1)
```
results in `99.8783`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.244 ODDLYIELD

Syntax:

```
ODDLYIELD ( settlement , maturity , last-interest , rate , pr , redemption ,
    frequency [ , [ basis ] ] )
```

Description: Computes the yield of a security that has an odd (short or long) last period.

Mathematical Formula:

$$ODDLYIELD = \left[\frac{\left(redemption + \left(\left(\sum_{i=1}^{NC}\frac{DC_i}{NL_i}\right)\times\frac{100\times rate}{frequency}\right)\right) - \left(par + \left(\left(\sum_{i=1}^{NC}\frac{A_i}{NL_i}\right)\times\frac{100\times rate}{frequency}\right)\right)}{pr + \left(\left(\sum_{i=1}^{NC}\frac{A_i}{NL_i}\right)\times\frac{100\times rate}{frequency}\right)}\right] \times \left[\frac{frequency}{\left(\sum_{i=1}^{NC}\frac{DSC_i}{NL_i}\right)}\right]$$

where:

- A_i = number of accrued days for the i^{th}, or last, quasi-coupon period within odd period counting forward from last interest date before redemption.
- DC_i = number of days counted in the i^{th}, or last, quasi-coupon period as delimited by the length of the actual coupon period.
- *frequency* = argument *frequency*
- *NC* = number of quasi-coupon periods that fit in odd period; if this number contains a fraction it is raised to the next whole number.
- NL_i = normal length in days of the i^{th}, or last, quasi-coupon period within odd coupon period.
- *par* = argument *par*
- *rate* = argument *rate*
- *redemption* = argument *redemption*
-

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
last-interest	number	The security's last coupon date.
rate	number	The security's interest rate.
pr	number	The security's price.
redemption	number	The security's redemption value per 100 currency units

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>face value.</td></tr>
<tr><td>frequency</td><td>number</td><td>the number of coupon payments per year. (For annual payments, frequency is 1; for semiannual payments, frequency is 2; for quarterly payments, frequency is 4.)</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days;</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		otherwise it is 365 days.
		2 — Actual/360. Similar to Basis 1, but only has 360 days per year.
		3 — Actual/365. Similar to Basis 1, but always has 365 days per year.
		4 — European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The yield of a security that has an odd (short or long) last period.

However, if

- *settlement*, *maturity*, or *last-interest* is out of range for the current date system, #NUM! is returned.
- The following is not true: *maturity* is later than *settlement*, which is later than *last-interest*, so #NUM! is returned.
- *rate* < 0, #NUM! is returned.
- *pr* ≤ 0, #NUM! is returned.
- *frequency* is any number other than 1, 2, or 4, #NUM! is returned.
- *basis* < 0 or *basis* > 4, #NUM! is returned.

[*Example*:

```
ODDLYIELD(DATE(2008,11,11),DATE(2021,3,1),DATE(2008,10,15),
  0.0575,84.5,100,2,0) results in 4.5192%
```

end example]

18.17.7.245 OFFSET

Syntax:

OFFSET (*reference* , *rows* , *cols* [, [*height*] [, [*width*]]])

Description: Gets a reference to a range that is a specified number of rows and columns from a cell or range of cells. The reference that is returned can be a single cell or a range of cells. You can specify the number of rows and the number of columns to be returned.

Arguments:

Name	Type	Description
reference	reference	Designates the base. *reference* shall refer to a cell or range of adjacent cells.
rows	number	The number of rows, up or down, that indicates the upper-left cell of the result to refer to. A positive value means below the starting reference; a negative value means above the starting reference.
cols	number	The number of columns, to the left or right, that the upper-left cell of the result to refer to. A positive value means to the right of the starting reference; a negative value means to the left of the starting reference.
height	number	The height, in rows, of the set of cells referred to by the resulting reference. This height shall be positive. If omitted, it is the same as the height of *reference*.
width	number	The width, in columns, of the set of cells referred to by the resulting reference. The width shall be positive. If omitted, it is the same as the width of *reference*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: reference – A reference to a range that is a specified size and number of rows and columns from a cell or range of cells.

However, if

- *reference* does not refer to a cell or range of adjacent cells, #VALUE! is returned.
- The combination of *rows* and *cols* results outside the worksheet, #REF! is returned.

[*Example*:

`OFFSET(C3,2,3,1,1)` results in the value in cell `F5`
`SUM(OFFSET(C3:E5,-1,0,3,3))` results in the sum of the range `C2:E4`

end example]

18.17.7.246 OR

Syntax:

`OR` (*argument-list*)

Description: Tests if any one or more *arguments* in *argument-list* are TRUE. The function evaluates all arguments prior to returning a value.

Arguments:

Name	Type	Description
argument-list	logical, array, reference.	The *arguments* in *argument-list* designate the values to be tested. For an array or cell reference, a cell that contains text or is empty shall be ignored.

Return Type and Value: logical – TRUE if any one or more *arguments* in *argument-list* are TRUE; otherwise, FALSE.

However, if no logical values are found, the return value is unspecified.

[*Example*:

`OR(TRUE)` results in TRUE
`OR(FALSE,FALSE)` results in FALSE
`OR(10=5,3=1+2,0)` results in TRUE
`OR({10,5,6,7},TRUE,E6:F6)` results in TRUE, when E6 contains FALSE and F6 contains 0

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.247 PEARSON

Syntax:

PEARSON (*array-1* , *array-2*)

Description: Computes the Pearson product moment correlation coefficient, a dimensionless index that ranges from -1.0 to 1.0, inclusive, and reflects the extent of a linear relationship between two data sets.

Mathematical Formula:

The formula for the Pearson product moment correlation coefficient, r, is:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*array-1*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*array-2*)

Arguments:

Name	Type	Description
array-1	number, name, array, reference to number	The set of independent numerical values. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
array-2	number, name, array, reference to number	The set of dependent numerical values. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The Pearson product moment correlation coefficient.

However, if

- *array-1* and *array-2* have a different number of data points, the return value is unspecified.
- *array-1* or *array-2* is empty, the return value is unspecified.

[*Example*:

`PEARSON({9,7,5,3,1},{10,6,1,5,3})` results in `0.699378606`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.248 PERCENTILE

Syntax:

```
PERCENTILE ( array , k )
```

Description: Computes the k^{th} percentile of a set of values in a range.

Arguments:

Name	Type	Description
array	array, reference	The set of numerical data that defines relative standing.
k	number	The percentile value in the range 0–1, inclusive. If *k* is not a multiple of 1/(n - 1), PERCENTILE interpolates to determine the value at the k^{th} percentile.

Return Type and Value: number – The k^{th} percentile of a set of values in a range.

However, if

- *array* is empty, the return value is unspecified.
- *k* is < 0 or *k* > 1, `#NUM!` is returned.

[*Example*:

`PERCENTILE({1,3,2,4},0.3)` results in `1.9`
`PERCENTILE({1,3,2,4},0.75)` results in `3.25`

end example]

18.17.7.249 PERCENTRANK

Syntax:

```
PERCENTRANK ( array , x [ , significance ] )
```

Description: Computes the rank of a value in a data set as a percentage of the data set.

Arguments:

Name	Type	Description
array	array, reference	*array* is the set of numerical data that defines relative standing.
x	number	The value for which the rank is to be computed. If *x* does

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		not match one of the values in *array*, PERCENTRANK interpolates to return the correct percentage rank.
significance	number	The number of significant digits for the returned percentage value. If omitted, a value of 3 is used.

Return Type and Value: number – The rank of a value in a data set as a percentage of the data set.

However, if

- *array* is empty, the return value is unspecified.
- *significance* < 1, `#NUM!` is returned.

[*Example*:

`PERCENTRANK({12,6,7,9,3,8},4)` results in `0.066`
`PERCENTRANK({12,6,7,9,3,8},5)` results in `0.133`

end example]

18.17.7.250 PERMUT

Syntax:

`PERMUT` (*number* , *number-chosen*)

Description: Computes the number of permutations for *number-chosen* objects that can be selected from *number* objects. [*Note*: A permutation is any set or subset of objects or events where internal order is significant. Permutations are different from combinations, for which the internal order is not significant. Use this function for lottery-style probability calculations. *end note*]

Mathematical Formula:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

where:

- k = argument *number-chosen*
- n = argument *number*

Arguments:

Name	Type	Description
number	number	The total number of items available, truncated to integer.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
number-chosen	number	The number of items in each permutation, truncated to integer.

Return Type and Value: number – The number of different permutations of *number-chosen* in *number*.

However, if

- *number* < 0, `#NUM!` is returned.
- *number-chosen* < 0, `#NUM!` is returned.
- *number* < *number-chosen*, `#NUM!` is returned.

[*Example*:

`PERMUT(8,2)` results in `56`
`PERMUT(10,4)` results in `5040`
`PERMUT(6,5)` results in `720`

end example]

18.17.7.251 PHONETIC

Syntax:

`PHONETIC ( string )`

Description: Extracts the phonetic (furigana) characters from *string*. [*Note*: Furigana are aids used to indicate correct pronunciation of Japanese text. *end note*]

Arguments:

Name	Type	Description
string	text, reference	Designates a furigana string. If *string* is a cell range, the furigana string in the upper-left corner cell of that range is returned.

Return Type and Value: text – The phonetic (furigana) characters from *string*.

However, if *string* is a range of non-contiguous cells, `#N/A` is returned.

18.17.7.252 PI

Syntax:

`PI ()`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Computes the value π.

Arguments: None.

Return Type and Value: number – The value π.

[*Example*: The following results are displayed using 10 significant digits

`PI()` results in `3.141592654`
`PI()/2` results in `1.570796327`
`PI()*(2.5^2)` results in `19.63495408`

end example]

18.17.7.253 PMT

Syntax:

`PMT ( rate , nper , pv [ , [ fv ] [ , [ type ] ] ] )`

Description: Computes the payment for a loan based on constant payments and a constant interest rate.

Arguments:

Name	Type	Description
rate	number	The interest rate for the loan.
nper	number	The total number of payment for the loan.
pv	number	The present value, or the total amount that a series of future payments is worth now; also known as the principal.
fv	number	The future value, or a cash balance to be attained after the last payment is made. If omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0).
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
`0`	Payment at the end of the period
`1`	Payment at the beginning of the period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The payment for a loan based on constant payments and a constant interest rate. (The payment returned by PMT includes principal and interest but no taxes, reserve payments, or fees sometimes associated with loans.)

Howver, if *type* is any number other than 0 or 1, `#NUM!` is returned.

[*Example*:

`PMT(0.08/12,10,10000)` results in `-1,037.03`
`PMT(0.08/12,10,10000,0,1)` results in `-1,030.16`

end example]

18.17.7.254 POISSON

Syntax:

`POISSON ( x , mean , cumulative-flag )`

Description: Computes the Poisson distribution.

Mathematical Formula:

For *cumulative-flag* = FALSE:

$$POISSON = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

For *cumulative-flag* = TRUE:

$$CUMPOISSON = \sum_{k=0}^{x} \frac{e^{-\lambda}\lambda^{k}}{k!}$$

where:

- x = argument *x*
- λ = argument *mean*

Arguments:

Name	Type	Description
x	number	The number of events, truncated to an integer.
mean	number	The expected numeric value.
cumulative-flag	logical	Determines the form of the function. If TRUE, POISSON returns the cumulative Poisson probability that the number of random events occurring are between zero and *x*, inclusive; if FALSE, it returns the Poisson

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		probability mass function that the number of events occurring is exactly x.

Return Type and Value: number – The Poisson distribution.

However, if

- $x < 0$, `#NUM!` is returned.
- *mean* ≤ 0, `#NUM!` is returned.

[*Example*:

`POISSON(2,5,TRUE)` results in `0.124652019`
`POISSON(2,5,FALSE)` results in `0.084224337`

end example]

18.17.7.255 POWER

Syntax:

```
POWER ( x , y )
```

Description: Computes x raised to the power y.

Arguments:

Name	Type	Description
x	number	The base and the number y is the exponent to which that base is raised.
y	number	The exponent to which the base is raised.

Return Type and Value: number – x^y.

However, if

- The value of x is negative and y is not a whole number, `#NUM!` is returned.
- x is zero and y is less than or equal to zero, `#DIV/0!` is returned.
- The result cannot be represented as a number, `#NUM!` is returned.

[*Example*:

`POWER(2,3)` results in 8

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`POWER(2,0.5)` results in `1.414213562`
`POWER(-1.234,5.0)` results in `-2.861381721`
`POWER(1.234,5.1)` results in `2.922182358`

end example]

18.17.7.256 PPMT

Syntax:

`PPMT ( rate , per , nper , pv [ , [ fv ][ , [ type ]]] )`

Description: Computes the payment on the principal for a given period for an investment based on periodic, constant payments and a constant interest rate.

Arguments:

Name	Type	Description
rate	number	The interest rate per period.
per	number	The the period and shall be in the range 1–*nper*.
nper	number	The total number of payment in an annuity.
pv	number	The present value, or the total amount that a series of future payments is worth now.
fv	number	The future value, or a cash balance to be attained after the last payment is made. If omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0).
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
`0`	Payment at the end of the period
`1`	Payment at the beginning of the period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The payment on the principal for a given period for an investment based on periodic, constant payments and a constant interest rate.

However, if *type* is any number other than 0 or 1, `#NUM!` is returned.

[*Example*:

`PPMT(0.1/12,1,2*12,2000)` results in `-75.62`
`PPMT(0.08,10,10,200000)` results in `-27,598.05`

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.257 PRICE

Syntax:

PRICE (*settlement* , *maturity* , *rate* , *yld* , *redemption* , *frequency* [, [*basis*]])

Description: Computes the price per 100 currency units face value of a security that pays periodic interest.

Mathematical Formula:

$$PRICE = \left[\frac{reemption}{\left(1+\frac{yld}{frequency}\right)^{\left(N-1+\frac{DSC}{E}\right)}}\right] + \left[\sum_{k=1}^{N}\frac{100\times\frac{rate}{frequency}}{\left(1+\frac{yld}{frequency}\right)^{\left(k-1+\frac{DSC}{E}\right)}}\right] - \left(100\times\frac{rate}{frequency}\times\frac{A}{E}\right)$$

where:

- A = number of days from beginning of coupon period to settlement date.
- DSC = number of days from settlement to next coupon date.
- E = number of days in coupon period in which the settlement date falls.
- *frequency* = argument *frequency*
- N = number of coupons payable between settlement date and redemption date.
- *rate* = argument *rate*
- *redemption* = argument *redemption*
- *yld* = argument *yld*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
rate	number	The security's interest rate.
yield	number	The security's annual yield.
redemption	number	The security's redemption value per 100 currency units face value.
frequency	number	the number of coupon payments per year. (For annual payments, *frequency* is 1; for semiannual payments, *frequency* is 2; for quarterly payments, *frequency* is 4.) *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description

Value	Day Count Basis
0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
2	Actual/360. Similar to Basis 1, but only has 360 days per year.
3	Actual/365. Similar to Basis 1, but always has

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>365 days per year.</td></tr><tr><td>4</td><td>European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: - If the date is 28 or 29 February, it is adjusted to 30 February. - For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February. </td></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The price per 100 currency units face value of a security that pays periodic interest.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *rate* or *yld* < 0, `#NUM!` is returned.
- *redemption* ≤ 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.

 ©ISO/IEC 2016 – All rights reserved

- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`PRICE(DATE(2008,2,15),DATE(2017,11,15),0.0575,0.065,100,2,0)` results in `94.6344`

end example]

18.17.7.258 PRICEDISC

Syntax:

`PRICEDISC ( `*settlement*` , `*maturity*` , `*discount*` , `*redemption*` [ , [ `*basis*` ] ] )`

Description: Computes the price per 100 currency units face value of a discounted security.

Mathematical Formula:

$$PRICEDISC = redemption - discount \times redemption \times \frac{DSM}{B}$$

where:

- B = number of days in year, depending on year basis.
- DSM = number of days from settlement to maturity.
- *discount* = argument *discount*
- *redemption* = argument *redemption*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
discount	number	The security's discount rate.
redemption	number	The security's redemption value per 100 currency units face value.
basis	number	The truncated integer type of day count basis to use, as follows: **Value** / **Day Count Basis** `0` or omitted / US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Name	Type	Description		
				following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
			1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
			2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			3	Actual/365. Similar to Basis 1, but always has 365 days per year.
			4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
			days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Time information in the date arguments is ignored.

Return Type and Value: number – The price per 100 currency units face value of a discounted security.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *discount* or *redemption* ≤ 0, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`PRICEDISC(DATE(2008,2,16),DATE(2008,3,1),0.0525,100,2)` results in `99.7958`

end example]

©ISO/IEC 2016 – All rights reserved

18.17.7.259 PRICEMAT

Syntax:

PRICEMAT (*settlement* , *maturity* , *issue* , *rate* , *yld* [, [*basis*]])

Description: Computes the price per 100 currency units face value of a security that pays interest at maturity.

Mathematical Formula:

$$PRICEMAT = \frac{100 + \left(\frac{IM}{B} \times rate \times 100\right)}{1 + \left(\frac{DSM}{B} \times yld\right)} - \left(\frac{A}{B} \times rate \times 100\right)$$

where:

- A = number of days from issue to settlement.
- B = number of days in year, depending on year basis.
- DIM = number of days from issue to maturity.
- DSM = number of days from settlement to maturity.
- *rate* = argument *rate*
- *yld* = argument *yld*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
issue	number	The security's issue date.
rate	number	The security's interest rate.
yld	number	The security's annual yield.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description		
				30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
			1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
			2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			3	Actual/365. Similar to Basis 1, but always has 365 days per year.
			4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

Return Type and Value: number – The price per 100 currency units face value of a security that pays interest at maturity.

However, if

- *settlement*, *maturity*, or *issue* is out of range for the current date system, #NUM! is returned.
- *settlement* ≥ *maturity*, #NUM! is returned.
- *rate* or *yld* < 0, #NUM! is returned.
- *basis* < 0 or *basis* > 4, #NUM! is returned.

[*Example*:

```
PRICEMAT(DATE(2008,2,15),DATE(2008,4,13),DATE(2007,11,11),0.061,0.061,0)
```

results in `99.9845`

end example]

18.17.7.260 PROB

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

PROB (*x-range* , *probability-range* , *lower-limit* [, *upper-limit*])

Description: Computes the probability that values in a range are between two limits.

Arguments:

Name	Type	Description
x-range	array, reference	The set of numeric values of *x* with which there are associated probabilities.
probability-range	array, reference	A set of numeric probabilities associated with the values in *x-range*.
lower-limit	number	The lower bound on the value for which the probability is to be computed.
upper-limit	number	The upper bound on the value for which the probability is to be computed. If omitted, the probability that values in *x-range* are equal to *lower-limit* is returned.

Return Type and Value: number – The probability that values in a range are between two limits.

However, if

- Any value in *probability-range* ≤ 0, `#NUM!` is returned.
- The sum of the values in *probability-range* <> 1, `#NUM!` is returned.
- *x-range* and *probability-range* contain a different number of data points, the return value is unspecified.

[*Example*:

`PROB({0,1,2,3},{0.2,0.3,0.1,0.4},2)` results in `0.1`
`PROB({0,1,2,3},{0.2,0.3,0.1,0.4},1,4)` results in `0.8`

end example]

18.17.7.261 PRODUCT

Syntax:

PRODUCT (*argument-list*)

Description: Multiplies the numeric values of *arguments* in *argument-list*.

Arguments:

Name	Type	Description
argument-list	logical, number, text, array, reference	The *arguments* in *argument-list* designate the numbers to be multiplied. Arguments that are numbers, logical values, or text representations of numbers shall be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		counted. If an argument is an array or reference, only numbers in that array or reference shall be counted. Empty cells, logical values, and text in the array or reference shall be ignored.

Return Type and Value: number – The product of the values of its arguments.

[*Example*:

`PRODUCT(1)` results in `1`
`PRODUCT(1,2,3,4,5)` results in `120`
`PRODUCT({1,2;3,4})` results in `24`
`PRODUCT({2,3},4,"5")` results in `120`

end example]

18.17.7.262 PROPER

Syntax:

PROPER (*string*)

Description: Makes a lowercase version of *string* except that the first letter in *string* and any other letters in *string* that immediately follow a character that is not a letter, are converted to uppercase.

Arguments:

Name	Type	Description
string	text, reference	Designates the string to be converted.

Return Type and Value: text – A version of *string* such that the first letter in *string* and any other letters in *string* that immediately follow a character that is not a letter, are converted to uppercase. All other letters are converted to lowercase, and all other non-letters are unchanged.

[*Example*:

`PROPER("12aBC d123aD#$%sd^")` results in `12Abc D123Ad#$%Sd^`
`PROPER(A10)` results in `12Abc D123Ad#$%Sd^`, when A10 contains `12aBC d123aD#$%sd^`

end example]

18.17.7.263 PV

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

PV (*rate* , *nper* , *pmt* [, [*fv*][, [*type*]]])

Description: Computes the present value of an investment. (The present value is the total amount that a series of future payments is worth now.)

Mathematical Formula:

If *rate* is not 0, then:

$$pv * (1 + rate)^{nper} + pmt(1 + rate * type) * \left(\frac{(1 + rate)^{nper} - 1}{rate}\right) + fv = 0$$

If *rate* is 0, then:

$(\text{pmt} \times \text{nper}) + \text{pv} + \text{fv} = 0$(pmt * nper) + pv + fv = 0

where:

- *fv* = *fv* argument
- *nper* = *nper* argument
- *pmt* = *pmt* argument
- *rate* = *rate* argument
- *type* = *type* argument

Arguments:

Name	Type	Description
rate	number	The interest rate per period.
nper	number	The total number of payment in an annuity.
pmt	number	The payment made each period and cannot change over the life of the annuity. If is omitted, *fv* shall be provided. [*Note*: Typically, *pmt* includes principal and interest but no other fees or taxes. *end note*]
fv	number	The future value, or a cash balance to be attained after the last payment is made. If omitted, *pmt* shall be provided.
type	number	The timing of the payment, truncated to integer, as follows:

Value	Timing
0	Payment at the end of the period
1	Payment at the beginning of the period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The present value of an investment.

However, if *type* is any number other than 0 or 1, `#NUM!` is returned.

[*Example*:

`PV(0.08/12,12*20,500,,0)` results in `-59,777.15`

end example]

18.17.7.264 QUARTILE

Syntax:

`QUARTILE ( array , result-category )`

Description: Computes the quartile of a data set.

Arguments:

Name	Type	Description
array	array, reference	The set of numeric values for which the quartile value is to be computed.
result-category	number	When truncated to an integer, specifies which value is to be returned, as follows: 0 – Minimum value 1 – First quartile (25th percentile) 2 – Median value (50th percentile) 3 – Third quartile (75th percentile) 4 – Maximum value

Return Type and Value: number – The quartile of a data set.

However, if

- *array* is empty, the return value is unspecified.
- *result-category* < 0 or *result-category* > 4, `#NUM!` is returned.

[*Example*:

`QUARTILE({1,2,4,7,8,9,10,12},1)` results in `3.5`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.265 QUOTIENT

Syntax:

```
QUOTIENT ( dividend , divisor )
```

Description: Computes the integer portion of the division of *dividend* by *divisor*.

Arguments:

Name	Type	Description
dividend	number	The dividend
divisor	number	The divisor.

Return Type and Value: number – The integer portion of the division of *dividend* by *divisor*.

[*Example*:

`QUOTIENT(5,2)` results in 2
`QUOTIENT(4.5,3.1)` results in `1`
`QUOTIENT(-10,3)` results in `-3`

end example]

18.17.7.266 RADIANS

Syntax:

```
RADIANS ( angle )
```

Description: Converts *angle* in degrees into radians.

Arguments:

Name	Type	Description
angle	number	The angle expressed in degrees that is to be converted into radians.

Return Type and Value: number – *angle* in radians.

[*Example*:

`RADIANS(360)` results in `6.283185307`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

`RADIANS(270)` results in `4.71238898`
`RADIANS(45)` results in `0.785398163`
`RADIANS(8.5)` results in `0.148352986`

end example]

18.17.7.267 RAND

Syntax:

```
RAND ( )
```

Description: Computes an evenly distributed random real number greater than or equal to 0 and less than 1. A new random real number is returned every time the cell's value is calculated.

Arguments: None.

Return Type and Value: number – An evenly distributed random real number greater than or equal to 0 and less than 1.

[*Example*:

`RAND()` results in `0.437337454`
`INT(RAND()*(6-1)+1)` might result in 3

end example]

18.17.7.268 RANDBETWEEN

Syntax:

```
RANDBETWEEN ( lower-bound , upper-bound )
```

Description: Computes a random integer number in the range *lower-bound*–*upper-bound*. A new random integer number is returned every time the cell's value is calculated.

Arguments:

Name	Type	Description
lower-bound	number	The smallest integer that is returned.
upper-bound	number	The largest integer that is returned.

Return Type and Value: number – A random integer number in the range specified.

However, if *lower-bound* is greater than *upper-bound*, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`RANDBETWEEN(1,6)` results in an integer between `1` and `6`, inclusive
`RANDBETWEEN(-10,10)` results in an integer between `-10` and `10`, inclusive

end example]

18.17.7.269 RANK

Syntax:

RANK (*number* , *number-list* [, *order*])

Description: Computes the rank of a number in a list of numbers. RANK gives duplicate numbers the same rank. However, the presence of duplicate numbers affects the ranks of subsequent numbers.

Arguments:

Name	Type	Description
number	number	The number whose rank is to be found.
number-list	reference	Designates the list of numbers. Non-numeric values in this list are ignored.
order	number	Specifies how *number* is to be ranked. If zero or omitted, *number* is ranked as if the list were sorted in descending order. If *order* is any non-zero value, *number* is ranked as if the list were sorted in ascending order.

Return Type and Value: number – The rank of a number in a list of numbers.

However, if *number* does not exist in *number-list*, #N/A is returned.

[*Example*:

When the cells `E1:I1` contain 7, 3.5, 3.5, 1, and 2

`RANK(E2,E1:I1,1)` results in 3
`RANK(E2,E1:I1,1,0)` results in 2
`RANK(E2,E1:I1,1)` results in 2
`RANK(E1,E1:I1,1)` results in 5, as the two 3.5 values both have a rank of 3; no value has rank 4.

end example]

18.17.7.270 RATE

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

RATE (*nper* , *pmt* , *pv* [, [[*fv*] [, [[*type*][, [*guess*]]]]]])

Description: Computes the interest rate per period of an annuity, using iteration, which can result in zero or more solutions.

Arguments:

Name	Type	Description
nper	number	The total number of payment periods.
pmt	number	The payment made each period and cannot change over the life of the annuity. (Typically, *pmt* includes principal and interest but no other fees or taxes.) If omitted, *fv* shall be present.
pv	number	The present value.
fv	number	The future value, or a cash balance to be attained after the last payment is made. If omitted, it is assumed to be 0 (i.e., the future value of a loan, for example, is 0).
type	number	The timing of the payment, truncated to integer, as follows: `0` – Payment at the end of the period `1` – Payment at the beginning of the period
guess	number	A guess for what the rate are. If omitted, it is assumed to be 10 percent.

Return Type and Value: number – The interest rate per period of an annuity.

However, if

- *type* is any number other than 0 or 1, `#NUM!` is returned.
- The result has not converged after an implementation-defined number of iterations, `#NUM!` is returned.

[*Example*:

`RATE(4*12,-200,8000)` results in `0.7701%`
`RATE(4*12,-200,8000)*12` results in `9.2418%`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.271 RECEIVED

Syntax:

RECEIVED (*settlement* , *maturity* , *investment* , *discount* [, [*basis*]])

Description: Computes the amount received at maturity for a fully invested security.

Mathematical Formula:

$$RECEIVED = \frac{investment}{1 - \left(discount \times \frac{DIM}{B}\right)}$$

where:

- B = number of days in a year, depending on the year basis.
- DIM = number of days from issue to maturity.
- *discount* = argument *discount*
- *investment* = argument *investment*

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
investment	number	The amount invested in the security.
discount	number	The security's discount rate.
basis	number	The truncated integer type of day count basis to use, as follows: **Value** / **Day Count Basis** 0 or omitted / US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
			has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

Return Type and Value: number – The amount received at maturity for a fully invested security.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *investment* or *discount* ≤ 0, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`RECEIVED(DATE(2008,2,15),DATE(2008,5,15),1000000,0.0575,2)` results in `1014584.65`

end example]

18.17.7.272 REPLACE

Syntax:

`REPLACE` (*string-1* , *start-pos* , *number-chars* , *string-2*)

Description: Produces a new string that is *string-1* with *number-chars* characters starting at position *start-pos*, replaced by *string-2*. (REPLACE is intended for use with languages that use the single-byte character set (SBCS), whereas REPLACEB (§18.17.7.273) is intended for use with languages that use the double-byte character set (DBCS).)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Arguments:

Name	Type	Description
string-1	text	Designates a string.
start-pos	number	The number of the start position within *string-1* from which characters in *string-1* are to be replaced. The start position of the first character is 1. *start-pos* shall be at least 1. If *start-pos* is beyond the end of *string-1*, the result is a new string that is *string-2* appended to *string-1*. If *start-pos* is within the bounds of *string-1*, but *number-chars* goes beyond the end of *string-1*, the characters starting at position *start-pos* through to the end of *string-1* shall be replaced by *string-2*.
number-chars	number	The number of characters within *string-1* that are to be replaced by the string designated by *string-2*.
string-2	text	Designates a string.

Return Type and Value: text – A copy of *string-1* with replacement characters from *string-2*.

However, if

- *start-pos* < 0, #VALUE! is returned.
- *number-chars* < 0, #VALUE! is returned.

[*Example*:

`REPLACE("abcdefghijk",3,4,"XY")` results in `abXYghijk`
`REPLACE("abcdefghijk",3,1,"12345")` results in `ab12345defghijk`
`REPLACE("abcdefghijk",15,4,"XY")` results in `abcdefghijkXY`

end example]

18.17.7.273 REPLACEB

Syntax:

`REPLACEB ( string-1 , start-pos , number-bytes , string-2 )`

Description: Produces a new string that is *string-1* with *number-bytes* bytes starting at position *start-pos*, replaced by *string-2*. (REPLACEB is intended for use with languages that use the double-byte character set (DBCS), whereas REPLACE (§18.17.7.272) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
string-1	text	Designates a string.
start-pos	number	The number of the start position within *string-1* from which characters in *string-1* are to be replaced. The start position of the first character is 1. *start-pos* shall be at least 0. If *start-pos* is beyond the end of *string-1*, the result is a new string that is *string-2* appended to *string-1*. If *start-pos* is within the bounds of *string-1*, but *number- bytes* goes beyond the end of *string-1*, the characters starting at position *start-pos* through to the end of *string-1* shall be replaced by *string-2*.
number-bytes	number	The number of characters within *string-1* that are to be replaced by the string designated by *string-2*.
string-2	text	Designates a string.

Return Type and Value: text – A copy of *string-1* with replacement characters from *string-2*.

However, if

- *start-pos* < 0, #VALUE! is returned.
- *number-bytes* < 0, #VALUE! is returned.

[*Example*: Assuming 1-byte characters:

`REPLACEB("abcdefghijk",3,4,"XY")` results in `abXYghijk`
`REPLACEB("abcdefghijk",3,1,"12345")` results in `ab12345defghijk`
`REPLACEB("abcdefghijk",15,4,"XY")` results in `abcdefghijkXY`

end example]

18.17.7.274 REPT

Syntax:

`REPT` (*string* , *replication-count*)

Description: Creates a string that is *replication-count* number of occurrences of *string* concatenated together.

Arguments:

Name	Type	Description
string	text	Designate the string to be replicated.
replication-count	number	The number of times *string* is to be replicated, truncated to integer. If *replication-count* is 0, the resulting string is empty.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: text – The final replicated string.

However, if *replication-count* < 0, #VALUE! is returned.

[*Example*:

`REPT("ABC",3)` results in ABCABCABC
`LEN(REPT("ABC",0))` results in `0`

end example]

18.17.7.275 RIGHT

Syntax:

`RIGHT ( string [ , number-chars ] )`

Description: Extracts the right-most *number-chars* characters from *string*. RIGHT is intended for use with languages that use the single-byte character set (SBCS), whereas RIGHTB (§18.17.7.276) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-chars	number	The number of characters to be extracted. If omitted, a count of 1 shall be assumed. *number-chars* shall be at least 0. If *number-chars* exceeds the length of *string*, the whole of *string* shall be extracted.

Return Type and Value: text – A string containing the right-most *number-chars* characters from *string*.

However, if *number-chars* < 0, #VALUE! is returned.

[*Example*:

`RIGHT("abcdef",2)` results in `ef`
`RIGHT(A10,4)` results in `z123`, when A10 contains `xyz123`

end example]

18.17.7.276 RIGHTB

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

RIGHTB (*string* , [*number-bytes*])

Description: Extracts the right-most *number-bytes*-worth of characters from *string*. (RIGHTB is intended for use with languages that use the double-byte character set (DBCS), whereas RIGHT (§18.17.7.275) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

Name	Type	Description
string	text	Designate the string from which a substring is to be extracted.
number-bytes	number	The number of bytes to be extracted. If omitted, a count of 1 shall be assumed. *number-bytes* shall be at least 0. If *number-bytes* exceeds the length of *string*, the whole of *string* shall be extracted.

Return Type and Value: text – A string containing the right-most *number-bytes*-worth of characters from *string*.

However, if *number-bytes* < 0, #VALUE! is returned.

[*Example*: Assuming 1-byte characters:

RIGHTB("abcdef",2) results in ef
RIGHTB(A10,4) results in z123, when A10 contains xyz123

end example]

18.17.7.277 ROMAN

Syntax:

ROMAN (*number* , *form*)

Description: Converts the Arabic number, *number*, to a Roman number according to *form*.

Arguments:

Name	Type	Description
number	number	The Arabic number to be converted.
form	number	Specifies the type of Roman numeral to be produced. The Roman numeral style ranges from Classic to Simplified, becoming more concise as the value of *form* increases, as follows: **Value** | **Type**

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		0, omitted, or TRUE: Classic. Only subtract powers of ten (but not L or V). Do not subtract a number from one that is more than 10 times greater. If another letter follows the larger one, it shall be smaller than the number preceding the larger one. 1: Concise. Allow subtractraction of L and V as well as powers of ten. Do not subtract a number from one that is more than 10 times greater. If another letter follows the larger one, it shall be smaller than the number preceding the larger one. 2: More concise. Allow subtractraction of L (but not V) as well as powers of ten. Allow subtraction of a number from one that is more than 10 times greater. If another letter follows the larger one, it shall be smaller than the number preceding the larger one. 3: Most concise. Allow subtractraction of L and V as well as powers of ten. Allow subtraction of a number from one that is more than 10 times greater. If another letter follows the larger one, it shall be smaller than the number preceding the larger one. 4 or FALSE: Simplified. Produce the fewest Roman digits.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: text – The corresponding Roman number.

However, if

- *number* < 0 or > 3999, #VALUE! is returned.
- *form* is not one of the values listed above, #VALUE! is returned.

[*Example*:

`ROMAN(499,0)` results in `CDXCIX`, which is 100 less than 500, plus 10 less than 100, plus one less than 10.
`ROMAN(499,1)` results in `LDVLIV`, which is 50 less than 500, plus 5 less than 50, plus one less than 5.
`ROMAN(499,2)` results in `XDIX`, which is 10 less than 500, plus one less than 10.
`ROMAN(499,3)` results in `VDIV`, which is 5 less than 500, plus one less than 5.

 ©ISO/IEC 2016 – All rights reserved

`ROMAN(499,4)` results in `ID`, which is 1 less than 500.
`ROMAN(2013,0)` results in `MMXIII`, which is 2,000, plus 10, plus 3.

end example]

18.17.7.278 ROUND

Syntax:

ROUND (*x* , *number-digits*)

Description: Rounds *x* to the number of digits specified by *number-digits*.

Arguments:

Name	Type	Description
x	number	The value to be rounded.
number-digits	number	The number of digits to which *x* is to be rounded. If *number-digits* is greater than 0, *x* is rounded to the specified number of decimal places. If *number-digits* is 0, *x* is rounded to the nearest integer. If *number-digits* is less than 0, *x* is rounded to the left of the decimal point.

Return Type and Value: number – The rounded value of *x*. When rounding, digits 0–4 round down, while digits 5–9 round up.

[*Example*:

`ROUND(2.15,1)` results in `2.2`
`ROUND(2.149,1)` results in `2.1`
`ROUND(-1.475,2)` results in `-1.48`
`ROUND(21.5,-1)` results in `20`

end example]

18.17.7.279 ROUNDDOWN

Syntax:

ROUNDDOWN (*x* , *number-digits*)

Description: Computes *x* rounded down, toward zero, to the number of digits specified by *number-digits*. [*Note*: ROUNDDOWN behaves like ROUND (§18.17.7.278), except that ROUNDDOWN always rounds a number down. *end note*]

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
x	number	The value to be rounded down.
number-digits	number	The number of digits to which *x* is to be rounded. If *number-digits* is greater than 0, *x* is rounded to the specified number of decimal places. If *number-digits* is 0, *x* is rounded to the nearest integer. If *number-digits* is less than 0, *x* is rounded to the left of the decimal point.

Return Type and Value: number – The rounded-down value of *x*.

[*Example*:

`ROUNDDOWN(3.2,0)` rounds 3.2 down to zero decimal places; that is, to 3
`ROUNDDOWN(76.9,0)` rounds 76.9 down to zero decimal places; that is, to 76
`ROUNDDOWN(3.14159,3)` rounds 3.14159 down to three decimal places; that is, to `3.141`
`ROUNDDOWN(-3.14159,1)` rounds -3.14159 down to one decimal place; that is, to `-3.1`
`ROUNDDOWN(31415.92654,-2)` rounds 31415.92654 down to two decimal places to the left of the decimal; that is, to `31400`

end example]

18.17.7.280 ROUNDUP

Syntax:

ROUNDUP (*x* , *number-digits*)

Description: Computes *x* rounded up, away from zero, to the number of digits specified by *number-digits*. [*Note*: ROUNDUP behaves like ROUND (§18.17.7.278), except that ROUNDUP always rounds a number up. *end note*]

Arguments:

Name	Type	Description
x	number	The value to be rounded up.
number-digits	number	The number of digits to which *x* is to be rounded. If *number-digits* is greater than 0, *x* is rounded up to the specified number of decimal places. If *number-digits* is 0, *x* is rounded up to the nearest integer. If *number-digits* is less than 0, *x* is rounded up to the left of the decimal point.

Return Type and Value: number – The rounded-up value of *x*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`ROUNDDOWN(3.2,0)` rounds 3.2 down to zero decimal places; that is, to 4
`ROUNDDOWN(76.9,0)` rounds 76.9 down to zero decimal places; that is, to 77
`ROUNDDOWN(3.14159,3)` rounds 3.14159 down to three decimal places; that is, to `3.142`
`ROUNDDOWN(-3.14159,1)` rounds -3.14159 down to one decimal place; that is, to `-3.2`
`ROUNDDOWN(31415.92654,-2)` rounds 31415.92654 down to two decimal places to the left of the decimal; that is, to `31500`

end example]

18.17.7.281 ROW

Syntax:

`ROW ( [ reference ] )`

Description: Finds the number of the row(s) corresponding to *reference*.

Arguments:

Name	Type	Description
reference	reference to a single cell or to a range of contiguous cells	If omitted, the behavior is as if *reference* referred to the cell containing the formula.

Return Type and Value: number – If *reference* refers to a single cell or to a single row of cells, the corresponding row is returned. If *reference* refers to a range of cells involving multiple rows, a vertical array of the corresponding rows as numbers is returned.

However, if the range of cells referred to by *reference* is not contiguous, `#REF!` is returned.

[*Example*:

`ROW()` results in 16, when the cell containing the formula is in row 16
`ROW(E17:G17)` results in 17
`ROW(E16:G17)` results in a vertical array containing 16 and 17, respectively

end example]

18.17.7.282 ROWS

Syntax:

`ROWS ( array )`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Finds the number of rows corresponding to *array*.

Arguments:

Name	Type	Description
array	array, reference to a single cell, or a reference to a range of contiguous cells	A set of rows.

Return Type and Value: number – The number of rows corresponding to *array*.

However, if the range of cells referred to by *array* is not contiguous, `#NULL!` is returned.

[*Example*:

`ROWS(E16:H16)` results in 1
`ROWS(E16:G18)` results in 3
`ROWS({1,2;3,4})` results in 2

end example]

18.17.7.283 RSQ

Syntax:

`RSQ` (*known-ys* , *known-xs*)

Description: Computes the square of the Pearson product moment correlation coefficient through data points in known ys and known xs.

Mathematical Formula:

The equation for the Pearson product moment correlation coefficient, r, is:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*known-xs*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*known-ys*)

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
known-xs	number, name, array, or reference to number, text, logical	Designate a set of numeric data points. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
known-ys	number, name, array, or reference to number, text, logical	Designate a set of numeric data points. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The square of the Pearson product moment correlation coefficient.

However, if

- *known-ys* and *known-xs* are empty or have a different number of data points, the return value is unspecified.
- *known-ys* and *known-xs* contain only one data point, the return value is unspecified.

[*Example*:

`RSQ({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in `0.057950192`

end example]

18.17.7.284 RTD

Syntax:

RTD (*progID* , [*rtd-server*] , *argument-list*)

Description: Retrieves data from a program in real-time. Periodically, this function returns new values and causes recalculation of the expression containing the call to it.

Arguments:

Name	Type	Description
progID	text	The name of the program from which the data is to be retrieved.
rtd-server	text	An optional string that is specific to the program with which RTD is communicating.
argument list	any	The presence and meaning of each *argument* in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		argument-list is specific to the program with which RTD is communicating.

Return Type and Value: array – The set of values returned by the program with which RTD is communicating.

[*Example*: Consider a stockprice program that is called as follows:

```
RTD("stockprice.rtd","NASD","MSFT")
```

The result it returns—the price of the stock MSFT according to NASD—changes over time, often every few seconds.

The *rtd-server* program could also be written to accept multiple arguments, allowing calls like the following::

```
RTD("stockprice.rtd","NASD","MSFT","GOOG","AMZN")
```

where three stock values are requested.

end example]

18.17.7.285 SEARCH

Syntax:

SEARCH (*string-1* , *string-2* [, *start-pos*])

Description: Performs a case-insensitive search, using a lexical comparison, for the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. (SEARCH is intended for use with languages that use the single-byte character set (SBCS), whereas SEARCHB (§18.17.7.286) is intended for use with languages that use the double-byte character set (DBCS).)

Arguments:

Name	Type	Description
string-1	text	Designate the string to be searched for within the string designated by *string-2*. *string-1* can contain the following wildcard characters: question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for an actual question mark or asterisk, that character shall be preceded by a tilde (~).
string-2	text	
start-pos	number	The number of the start position within *string-2* for which *string-1* is to be searched. The start position of the first character is 1. If omitted, a position of 1 shall be assumed. *start-pos* shall be at least 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The start position of the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. If *string-1* is an empty string, it shall always be found in any *string-2* at position *start-pos*, or at position 1 if *start-pos* is omitted.

However, if

- *string-1* is not found within *string-2*, #VALUE! is returned.
- *start-pos* designates a position outside *string-2*, #VALUE! is returned.

[*Example*:

SEARCH("de","abcdEF") results in 4
SEARCH("?c*e","abcdEF") results in 2

end example]

18.17.7.286 SEARCHB

Syntax:

SEARCHB (*string-1* , *string-2* [, *start-pos*])

Description: Performs a case-insensitive search, using a lexical comparison, for the first occurrence of *string-1* in *string-2*, starting at byte position *start-pos* within *string-2*. (SEARCHB is intended for use with languages that use the double-byte character set (DBCS), whereas SEARCH (§18.17.7.285) is intended for use with languages that use the single-byte character set (SBCS).)

Arguments:

Name	Type	Description
string-1	text	Designate the string to be searched for within the string designated by *string-2*. *string-1* can contain the following wildcard characters: question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for an actual question mark or asterisk, that character shall be preceded by a tilde (~).
string-2	text	
start-pos	number	The number of the start position within *string-2* for which *string-1* is to be searched. The start position of the first byte is 1. If omitted, a position of 1 shall be assumed. *start-pos* shall be at least 0.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The start position of the first occurrence of *string-1* in *string-2*, starting at character position *start-pos* within *string-2*. If *string-1* is an empty string, it shall always be found in any *string-2* at position *start-pos*, or at position 1 if *start-pos* is omitted.

However, if

- *string-1* is not found within *string-2*, #VALUE! is returned.
- *start-pos* designates a position outside *string-2*, #VALUE! is returned.

[*Example*: Assuming 1-byte characters

SEARCHB("de","abcdEF") results in 4
SEARCHB("?c*e","abcdEF") results in 2

end example]

18.17.7.287 SECOND

Syntax:

SECOND (*time-value*)

Description: Computes the second for the date and/or time having the given *time-value*.

Arguments:

Name	Type	Description
time-value	number	The date and/or time whose second is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its integer part is ignored, or as a *string-constant* having any date and/or time format, in which case, any date information shall be ignored.

Return Type and Value: number – The second for the date and/or time having the given *time-value*.

However, if *time-value* is out of range for the current date system, #NUM! is returned.

[*Example*:

SECOND(DATE(2006,2,26)+TIME(2,10,20)) results in 20
SECOND(TIME(22,56,34)) results in 34
SECOND(0) results in 0, since serial date-time 0 represents 00:00:00
SECOND(10.5) results in 0, since serial date-time .5 represents 12:00:00
SECOND("22-Oct-2001 10:53:12") results in 12
SECOND("10:53:12 pm") results in 12

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`SECOND("22:53:12")` results in 12

end example]

18.17.7.288 SERIESSUM

Syntax:

`SERIESSUM` (*input-value* , *initial-power* , *step* , *coefficients*)

Description: Computes the sum of a power series.

Mathematical Formula:

The sum of a power series is based on the formula:

$$SERIES(x, n, m, a) = a_1 x^n + a_2 x^{(n+m)} + a_3 x^{(n+2m)} + \cdots + a_i x^{(n+(i-1)m)}$$

where:

- a = argument *coefficients*
- m = argument *step*
- n = argument *initial-power*
- x = argument *input-value*

Arguments:

Name	Type	Description
input-value	number	The input value to the power series;
initial-power	number	The initial power to which *input-value* is to be raised.
step	number	The step by which to increase *initial-power* for each term in the series;
coefficients	reference	A set of coefficients by which each successive power of *input-value* is multiplied. The number of values in *coefficients* determines the number of terms in the power series.

Return Type and Value: number – The sum of a power series.

[*Example*: Given the following data:

	A
1	1
2	=-1/FACT(2)
3	=1/FACT(4)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	A
4	=-1/FACT(6)

`SERIESSUM(PI()/4,0,2,A1:A4)` results in `0.707103`, an approximation to the cosine of π/4 radians

end example]

18.17.7.289 SIGN

Syntax:

```
SIGN ( x )
```

Description: Determines the sign of *x*.

Arguments:

Name	Type	Description
x	number	The number whose sign is to be determined.

Return Type and Value: number – 1 if *x* is positive, 0 if *x* is 0, and -1 if *x* is negative.

[*Example*:

`SIGN(10.5)` results in `1`
`SIGN(0)` results in `0`
`SIGN(-5.4)` results in `-1`

end example]

18.17.7.290 SIN

Syntax:

```
SIN ( x )
```

Description: Computes the sine of *x*.

Arguments:

Name	Type	Description
x	number	The number, in radians, whose sine is to be computed.

Return Type and Value: number – The sine of *x*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`SIN(-1)` results in `-0.841470985`
`SIN(0)` results in `0`
`SIN(1)` results in `0.841470985`

end example]

18.17.7.291 SINH

Syntax:

`SINH ( x )`

Description: Computes the hyperbolic sine of *x*.

Arguments:

Name	Type	Description
x	number	The number whose hyperbolic sine is to be computed.

Return Type and Value: number – The hyperbolic sine of *x*.

However, if the magnitude of *x* is too large for the result to be represented, `#NUM!` is returned.

[*Example*:|

`SINH(1)` results in `1.175201194`
`SINH(10)` results in `11013.23287`
`SINH(100)` results in `1.34406E+43`

end example]

18.17.7.292 SKEW

Syntax:

`SKEW ( argument-list )`

Description: Computes the skewness of a distribution. [*Note*: Skewness characterizes the degree of asymmetry of a distribution around its mean. Positive skewness indicates a distribution with an asymmetric tail extending toward more positive values. Negative skewness indicates a distribution with an asymmetric tail extending toward more negative values. *end note*]

Mathematical Formula:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

$$\frac{n}{(n-1)(n-2)}\sum\left(\frac{x_i-\bar{x}}{s}\right)^3$$

where:

- n = the number of elements in *argument-list*
- s = standard deviation of the values in *argument-list*
- xi = value of each element in *argument-list*
- $\bar{x}$ = mean of the values in *argument-list*

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *arguments* in *argument-list* designate the numbers for which the skewness is to be computed. Logical values and text representations of numbers that are entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The skewness of a distribution.

However, if

- There are fewer than three data points, the return value is unspecified.
- The sample standard deviation is zero, the return value is unspecified.

[*Example*:

SKEW(3,4,5,2,3,4,5,6,4,7) results in 0.359543071

end example]

18.17.7.293 SLN

SLN (*cost* , *salvage* , *life*)

Description: Computes the straight-line depreciation of an asset for one period.

Arguments:

Name	Type	Description
cost	number	The number *cost* is the initial cost of the asset.
salvage	number	The value at the end of the depreciation. (This is sometimes called the salvage value of the asset.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
life	number	The number of periods over which the asset is being depreciated. (This is sometimes called the useful life of the asset.)

Return Type and Value: number – The straight-line depreciation of an asset for one period.

[*Example*:

`SLN(30000,7500,10)` results in `2,250.00`

end example]

18.17.7.294 SLOPE

Syntax:

SLOPE (*known-ys* , *known-xs*)

Description: Computes the slope of the linear regression line through data points in known ys and known xs. The slope is the vertical distance divided by the horizontal distance between any two points on the line, which is the rate of change along the regression line.

Mathematical Formula:

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*known-xs*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*known-ys*)

Arguments:

Name	Type	Description
known-xs	number, name, array, or reference to number, text, logical	Designate a set of numeric dependent data points. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
known-ys	number, name, array, or	Designate a set of numeric independent data points. Logical values and text representations of numbers

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
	reference to number, text, logical	entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

Return Type and Value: number – The slope of the linear regression line through data points in known ys and known xs.

However, if

- *known-ys* and *known-xs* are empty, the return value is unspecified.
- *known-ys* and *known-xs* have a different number of data points, the return value is unspecified.

[*Example*:

`SLOPE({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in `0.305555556`

end example]

18.17.7.295 SMALL

Syntax:

`SMALL ( array , k )`

Description: Computes the k^{th} smallest value in a data set.

Arguments:

Name	Type	Description
array	array, reference	The set of numbers from which the k^{th}-smallest value is to be determined.
k	number	The position (from the smallest) in the array or cell range of data to return.

Return Type and Value: number – The k^{th} smallest value in a data set.

However, if

- *array* is empty, the return value is unspecified.
- $k \leq 0$ or k is greater than the number of data points, `#NUM!` is returned.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

SMALL({3,5,3,5,4;4,2,4,6,7},3) results in 3
SMALL({3,5,3,5,4;4,2,4,6,7},7) results in 5

end example]

18.17.7.296 SQRT

Syntax:

SQRT (*x*)

Description: Computes the positive square root of *x*.

Arguments:

Name	Type	Description
x	number	The number whose positive root is to be found.

Return Type and Value: number – The positive square root of *x*.

However, if *x* is negative, #NUM! is returned.

[*Example*:

SQRT(2) results in 1.414213562
SQRT(5) results in 2.236067977

end example]

18.17.7.297 SQRTPI

Syntax:

SQRTPI (*x*)

Description: Computes the positive square root of $x \times \pi$.

Arguments:

Name	Type	Description
x	number	The number, which when multiplied by π, whose positive root is to be found.

Return Type and Value: number – The positive square root of $x \times \pi$.

However, if *x* is negative, #NUM! is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*:

`SQRTPI(1)` results in `1.772453851`
`SQRTPI(2)` results in `2.506628275`

end example]

18.17.7.298 STANDARDIZE

Syntax:

`STANDARDIZE` (*x* , *mean* , *standard-dev*)

Description: Computes a normalized value from a distribution characterized by *mean* and *standard-dev*.

Mathematical Formula:

$$Z = \frac{X - \mu}{\sigma}$$

where:

- X = argument *x*
- μ = argument *mean*
- σ = argument *standard-dev*

Arguments:

Name	Type	Description
x	number	The number whose value is to be normalized. Represented by X in the mathematical formula presented.
mean	number	The arithmetic mean of the distribution. Represented by mu (μ) in the mathematical formula presented.
standard-dev	number	The standard deviation of the distribution. Represented by sigma (σ) in the mathematical formula presented.

Return Type and Value: number – A normalized value from a distribution.

However, if *standard-dev* ≤ 0, `#NUM!` is returned.

[*Example*:

`STANDARDIZE(42,40,1.5)` results in `1.333333333`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.299 STDEV

Syntax:

STDEV (*argument-list*)

Description: Makes an estimate of the standard deviation based on a sample, using the "unbiased" or "n-1" method. [*Note*: STDEV assumes that its arguments are a sample of the population. If the data represents the entire population, STDEVP should be used instead. If logical values and text representations of numbers in a reference are to be included as part of the calculation, use STDEVA instead. *end note*]

Mathematical Formula:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{(n-1)}}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-2*, …, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *arguments* in *argument-list* designate the numbers that are samples of the population. *argument-list* can also be an array of numbers. Logical values and text representations of numbers that are entered directly into the list of arguments are included. If an argument is an array or reference, only numbers in that array or reference are included. Empty cells, logical values, text, or error values in the array or reference are ignored.

Return Type and Value: number – An estimate of the standard deviation based on a sample.

However, if the sample size ≤ *1*, `#DIV/0!` is returned.

[*Example*:

`STDEV(123,134,143,173,112,109)` results in `23.72902583`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.300 STDEVA

Syntax:

STDEVA (*argument-list*)

Description: Makes an estimate of the standard deviation based on a sample, using the "unbiased" or "n-1" method. [*Note*: STDEVA assumes that its arguments are a sample of the population. If the data represents the entire population, STDEVPA should be used instead. If logical values and text representations of numbers in a reference are to be excluded as part of the calculation, use STDEV instead. *end note*]

Mathematical Formula:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{(n-1)}}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-2*, …, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference. The argument list can also be an array of numbers.	The *arguments* in *argument-list* designate the numbers that are samples of the population. Arguments that contain TRUE evaluate as 1; arguments that contain text or FALSE evaluate as zero. If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored.

Return Type and Value: number – An estimate of the standard deviation based on a sample.

However, if the sample size ≤ *1*, `#DIV/0!` is returned.

[*Example*:

`STDEVA(123,134,143,173,112,109)` results in `23.72902583`

end example]

18.17.7.301 STDEVP

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

STDEVP (*argument-list*)

Description: Computes the standard deviation of an entire population, using the "biased" or "n" method. [*Note*: STDEVP assumes that its arguments are the total population. If the data represents a population sample only, STDEVA should be used instead. If logical values and text representations of numbers in a reference are to be included as part of the calculation, use STDEVPA instead. *end note*]

Mathematical Formula:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{n}}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-2*, …, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference. The argument list can also be an array of numbers.	The *arguments* in *argument-list* designate the numbers that are the members of the population. Logical values, and text representations of numbers that are entered directly into the list of arguments are included. If an argument is an array or reference, only numbers in that array or reference are included. Empty cells, logical values, text, or error values in the array or reference are ignored.

Return Type and Value: number – The standard deviation of an entire population.

However, if the sample size is zero, `#DIV/0!` is returned.

[*Example*:

STDEVP(`123,134,143,173,112,109`) results in `21.66153785`

end example]

18.17.7.302 STDEVPA

Syntax:

STDEVPA (*argument-list*)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Computes the standard deviation of an entire population, using the "biased" or "n" method. [*Note*: STDEVPA assumes that its arguments are the total population. If the data represents a population sample only, STDEVA should be used instead. If logical values and text representations of numbers in a reference are to be excluded as part of the calculation, use STDEVP instead. *end note*]

Mathematical Formula:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{n}}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-2*, …, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference. The argument list can also be an array of numbers.	The *argument*s in *argument-list* designate the numbers that are the members of the population.

Return Type and Value: number – The standard deviation of an entire population.

Arguments can be numbers; names, arrays, or references that contain numbers; text representations of numbers; or logical values, in a reference. Text representations of numbers that are entered directly into the list of arguments are included. Arguments that contain TRUE evaluate as 1; arguments that contain text or FALSE evaluate as zero. If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored.

However, if the sample size is zero, `#DIV/0!` is returned.

[*Example*:

STDEVPA(`123,134,143,173,112,109`) results in `21.66153785`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.303 STEYX

Syntax:

STEYX (*known-ys* , *known-xs*)

Description: Computes the standard error of the predicted *y*-value for each *x* in the regression. The standard error is a measure of the amount of error in the prediction of *y* for an individual *x*.

Mathematical Formula:

$$\sqrt{\frac{1}{(n-2)}\left[\sum(y-\bar{y})^2 - \frac{[\sum(x-\bar{x})(y-\bar{y})]^2}{\sum(x-\bar{x})^2}\right]}$$

where:

- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*known-xs*)
- y = a sample value
- $\bar{y}$ = the sample mean AVERAGE(*known-ys*)

Arguments:

Name	Type	Description
known-xs	number, name, array, or reference to number, text, logical	Designate a set of numeric dependent data points. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.
known-ys	number, name, array, or reference to number, text, logical	Designate a set of numeric independent data points. Logical values and text representations of numbers entered directly into the list of arguments are included. If an array or reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value 0 are included.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The standard error of the predicted y-value for each *x* in the regression.

However, if

- *known-ys* and *known-xs* have a different number of data points, the return value is unspecified.
- *known-ys* and *known-xs* are empty or have less than three data points, the return value is unspecified.

[*Example*:

©ISO/IEC 2016 – All rights reserved

`STEYX({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in `3.30571895`

end example]

18.17.7.304 SUBSTITUTE

Syntax:

`SUBSTITUTE ( string , old-string , new-string [ , occurrence ] )`

Description: Produces a new string that is *string* with one or all occurrences of *old-string* replaced by *new-string*.

Arguments:

Name	Type	Description
string	text	Designates a string.
old-string	text	Designates a string.
new string	text	Designates a string.
occurrence	number	The occurrence number of the *old-string* characters within *string-1* that is to be replaced by the string designated by *new-string*. If omitted, all occurrences of *old-string* characters shall be replaced.

Return Type and Value: text – A string that is *string* with one or all occurrences of *old-string* replaced by *new-string*.

However, if *occurrence* ≤ 0, #VALUE ! is returned.

[*Example*:

`SUBSTITUTE("abcaaabca","a","xx")` results in `xxbcxxxxxxbcxx`
`SUBSTITUTE("abcaaabca","a","",10)` results in `bcbc`
`SUBSTITUTE("abcaaabca","a","xx",3)` results in `abcaxxabca`

end example]

18.17.7.305 SUBTOTAL

Syntax:

`SUBTOTAL ( function-number , argument-list )`

Description: Computes a value using the function designated by *function-number*, using the *argument*s in *argument-list*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
function-number	number	Indicates the function to be called, as shown in the table below.
argument-list	number	Each *argument* in *argument-list* is passed to the called function, in the order specified. That shall be no more than 254 *argument*s.

function-number (includes hidden values)	*function-number* (excludes hidden values)	Function
1	101	AVERAGE
2	102	COUNT
3	103	COUNTA
4	104	MAX
5	105	MIN
6	106	PRODUCT
7	107	STDEV
8	108	STDEVP
9	109	SUM
10	110	VAR
11	111	VARP

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

If any argument contains a SUBTOTAL function call, that call shall be ignored to avoid double counting.

For the *function-number* values 1–11, the values of hidden rows are included. For the *function-number* values 101–111, the values of hidden rows are excluded.

The SUBTOTAL function shall ignore any rows that are not included in the result of a filter, regardless of which *function-number* value is used.

The SUBTOTAL function is designed for columns of data, or vertical ranges. It is not designed for rows of data, or horizontal ranges. [*Example*: When a horizontal range is subtotaled using a *function-number* of 101 or greater, hiding a column does not affect the subtotal. However, hiding a row in a subtotal of a vertical range does affect the subtotal. *end example*]

Return Type and Value: number – The result from calling the function designated by *function-number*, using the *argument*s in *argument-list*.

However, if *function-number* does not have one of the values specified above, `#NUM!` is returned.

©ISO/IEC 2016 – All rights reserved

[*Example*:

`SUBTOTAL(2,E5:E15)` counts the number of values in the cell range E5:E15, including hidden values
`SUBTOTAL(4,E5:E15)` finds the maximum value of the values in the cell range E5: E15, including hidden values
`SUBTOTAL(106,E5:E15)` finds the product of the values in the cell range E5: E15, excluding hidden values

end example]

18.17.7.306 SUM

Syntax:

`SUM (` *argument-list* `)`

Description: Adds the numeric values of *argument*s in *argument-list*.

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference.	The *argument*s in *argument-list* designate the numeric values to be added. Arguments that are numbers, logical values, or text representations of numbers shall be counted. If an argument is an array or reference, only numbers in that array or reference shall be counted. Empty cells, logical values, and text in the array or reference shall be ignored.

Return Type and Value: number – The sum of the values of its arguments.

[*Example*:

`SUM(1,2,3,4,5)` results in 15
`SUM({1,2;3,4})` results in 10
`SUM({1,2,3,4,5},6,"7")` results in 28
`SUM({1,"2",TRUE,4})` results in 5, as the logical value and numeric text are ignored

end example]

18.17.7.307 SUMIF

Syntax:

`SUMIF (` *cell-range* `,` *selection-criteria* `[ ,` *sum-range* `] )`

Description: Applies selection criteria on the values in one range of cells and sums the values of the cells in a corresponding range.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
cell-range	reference	Designates the range of cells to be inspected.
selection-criteria	number, expression, reference, text	Defines which cells are counted. In the case of text, *selection-criteria* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. *selection-criteria* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~).
sum-range	reference	If present, *sum-range* designates the cells whose values are summed. In this case, *sum-range* does not have to have the same size and shape as *cell-range*. The actual cells that are added are determined by using the top, left cell in *sum-range* as the beginning cell, and then including cells that correspond in size and shape to *cell-range*. If omitted, *cell-range* also designates the cells whose values are summed.

Return Type and Value: number – The sum of the cells corresponding to those selected.

[*Example*: Given that A1, B1, C1, and D1, respectively, contain the values 3, 10, 7, and 10

`SUMIF(A1:D1,"=10")` results in `20`
`SUMIF(A1:D1,">5")` results in `27`
`SUMIF(A1:D1,"<>10")` results in `10`

Given that A2, B2, C2, and D2, respectively, contain the values `apples`, `melons`, `10`, and `15`
`SUMIF(A2:B2,"*es",C2:D2)` results in `10`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.17.7.308 SUMIFS

Syntax:

```
SUMIFS ( sum-range , cell-range-1 , selection-criteria-1
   [ , cell-range-2 , selection-criteria-2 [ , … ] ] )
```

Description: Adds the cells in a range that meet multiple criteria.

Arguments:

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
sum-range	reference	Designates the cells whose values are summed. In this case, *sum-range* does not have to have the same size and shape as *cell-range-1* through *cell-range-n*. The actual cells that are added are determined by using the top, left cell in *sum-range* as the beginning cell, and then including cells that correspond in size and shape to *cell-range-1* through *cell-range-n*. Each cell in *sum-range* is summed only if all of the corresponding criteria specified are true for that cell. Cells in *sum-range* that contain TRUE evaluate to 1; cells in sum-range that contain FALSE evaluate to 0.
cell-range-1	reference	Designates the first range of cells to be inspected.
selection-criteria-1	number, expression, reference, text	Specifies the criteria for the first range of cells that is counted. In the case of text, *selection-criteria-1* can consist of any comparison operator followed by the operand against which each cell's value is to be compared. *selection-criteria-1* can include one or more wildcard characters, question mark (?) and asterisk (*). A question mark matches any single character; an asterisk matches any sequence of characters. To search for a question mark, asterisk, or tilde character, prefix that character with a tilde (~).
cell-range-n	reference	The optional arguments *selection-criteria-2* through *selection-criteria-n* have corresponding arguments *cell-range-2* through *cell-range-n*, and have the same semantics as *selection-criteria-1* and *cell-range-1*, respectively.
selection-criteria-n	number, expression, reference, text	

Return Type and Value: number – The sum of the cells corresponding to those selected.

[*Example*: Given the following data:

	A	B	C	D
1	Sales Person	Tables	Chairs	Desks
2	Emilio	34	85	97
3	Julie	353	23	18
4	Hans	13	67	14
5	Frederique	0	98	0

`SUMIFS(B2:C5,A2:A5,"=Julie")` results in `353` (the sum of the number of tables and chairs sold by Julie)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

`SUMIFS(B2:B5,A2:A5,"=Julie",A2:A5,"=Hans")` results in `0` (the sum of the number of tables sold by Julie and Hans)

`SUMIFS(B2:B5,A3,"=Julie",A4,"=Hans")` results in 34 (the sum of the the number of tables sold by Julie and Hans)

`SUMIFS(B2:D5,A2:A5,"<>Emilio")` results in 768 (the sum of the number of tables, chairs, and desks sold by all sales persons except Emilio)

end example]

18.17.7.309 SUMPRODUCT

Syntax:

SUMPRODUCT (*argument-list*)

Description: Multiplies the corresponding elements in the array *arguments* in *argument-list*, and returns the sum of those products. An array element that is not numeric is treated as if it contained 0.

Arguments:

Name	Type	Description
argument-list	array of numbers	The *arguments* in *argument-list* designate the numeric values to be multiplied.

Return Type and Value: number – The sum of the products of the corresponding elements in the *arguments* in *argument-list*.

However, if the array arguments do not have the same dimensions, `#NUM!` is returned.

[*Example*:

`SUMPRODUCT({2,3})` results in 5
`SUMPRODUCT({2,3},{4,5})` results in 23
`SUMPRODUCT({2,3},{4,5},{2,2})` results in 46
`SUMPRODUCT({2,3;4,5},{2,2;3,4})` results in 42

end example]

18.17.7.310 SUMSQ

Syntax:

SUMSQ (*argument-list*)

Description: Adds the squares of *arguments* in *argument-list*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *argument*s in *argument-list* designate the values whose squares are to be summed. Arguments that are numbers, logical values, or text representations of numbers shall be counted. If an argument is an array or reference, only numbers in that array or reference shall be counted. Empty cells, logical values, and text in the array or reference shall be ignored.

Return Type and Value: number – The sum of the squares of its arguments.

[*Example*:

`SUMSQ(2)` results in 4
`SUMSQ(2.5,-3.6))` results in `19.21`
`SUMSQ({2.5,-3.6)},2.4)` results in `24.97`

end example]

18.17.7.311 SUMX2MY2

Syntax:

`SUMX2MY2 ( array-1 , array-2 )`

Description: Computes the sum of the difference of squares of the corresponding numerical elements in two arrays designated by *array-1* and *array-2*.

Mathematical Formula:

$$SUMX2MY2 = \sum (x^2 - y^2)$$

where:

- *array-1* contains the x values
- *array-2* contains the y values

Arguments:

Name	Type	Description
array-1	array, reference	Designated the arrays to be operated on. If an argument contains text, logical values, or empty cells, those elements shall be ignored; however, cells with the value 0 shall be included.
array-2		

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The sum of the difference of squares of the corresponding elements in two arrays designated by *array-1* and *array-2*.

However, if *array-1* and *array-2* have a different number of values, the return value is unspecified.

[*Example*:

`SUMX2MY2({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in 55
`SUMX2MY2({2,3,9;1,8,7},{6,5,11;7,5,4})` results in -64
end example]

18.17.7.312 SUMX2PY2

Syntax:

`SUMX2PY2 ( array-1 , array-2 )`

Description: Computes the sum of the sum of the squares of the corresponding numerical elements in two arrays designated by *array-1* and *array-2*.

Mathematical Formula:

$$SUMX2PY2 = \sum(x^2 + y^2)$$

where:

- *array-1* contains the x values
- *array-2* contains the y values

Arguments:

Name	Type	Description
array-1	array, reference	Designated the arrays to be operated on. If an argument contains text, logical values, or empty cells, those elements shall be ignored; however, cells with the value 0 shall be included.
array-2		

Return Type and Value: number – The sum of the sum of the squares of the corresponding elements in two arrays designated by *array-1* and *array-2*.

However, if *array-1* and *array-2* have a different number of values, the return value is unspecified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

`SUMX2PY2({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in 521
`SUMX2PY2({2,3,9;1,8,7},{6,5,11;7,5,4})` results in 480

end example]

18.17.7.313 SUMXMY2

Syntax:

`SUMXMY2` (*array-1* , *array-2*)

Description: Computes the sum of the squares of the difference between corresponding numerical elements in two arrays designated by *array-1* and *array-2*.

Mathematical Formula:

$$SUMXMY2 = \sum (x - y)^2$$

where:

- *array-1* contains the *x* values
- *array-2* contains the *y* values

Arguments:

Name	Type	Description
array-1	array, reference	Designated the arrays to be operated on. If an argument contains text, logical values, or empty cells, those elements shall be ignored; however, cells with the value 0 shall be included.
array-2		

Return Type and Value: number – The sum of the squares of the difference between the corresponding elements in two arrays designated by *array-1* and *array-2*.

However, if *array-1* and *array-2* have a different number of values, the return value is unspecified.

[*Example*:

`SUMXMY2({2,3,9,1,8,7,5},{6,5,11,7,5,4,4})` results in 79
`SUMXMY2({2,3,9;1,8,7},{6,5,11;7,5,4})` results in 78

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.17.7.314 SYD

SYD (*cost* , *salvage* , *life* , *per*)

Description: Computes the sum-of-years' digits depreciation of an asset for a specified period.

Mathematical Formula:

$$SYD = \frac{(cost - salvage) * (life - per + 1) * 2}{(life)(life + 1)}$$

where:

- *cost* = argument *cost*
- *life* = argument *life*
- *per* = argument *per*
- *salvage* = argument *salvage*

Arguments:

Name	Type	Description
cost	number	The initial cost of the asset.
salvage	number	The value at the end of the depreciation. (This is sometimes called the salvage value of the asset.)
life	number	The number of periods over which the asset is being depreciated. (This is sometimes called the useful life of the asset.)
per	number	The period and shall have the same units as *life*.

Return Type and Value: number – The sum-of-years' digits depreciation of an asset for a specified period.

[*Example*:

SYD(30000,7500,10,1) results in 4,090.91
SYD(30000,7500,10,10) results in 409.09

end example]

18.17.7.315 T

Syntax:

T (*value*)

Description: Retrieves the text referencedby *value*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
value	any	The value to be tested for text. No conversion to text shall take place on an argument passed to this function.

Return Type and Value: text – *value* if *value* designates text; otherwise, "". [*Note*: T cannot differentiate between text that is an empty string, and any *value* of non-text type. *end note*]

[*Example*:

`T("Hello")` results in `Hello`
`T(123)` results in an empty string
`LEN(T(123))` results in `0`

end example]

18.17.7.316 TAN

Syntax:

TAN (*x*)

Description: Computes the tangent of *x*.

Arguments:

Name	Type	Description
x	number	The number, in radians, whose tangent is to be computed.

Return Type and Value: number – The tangent of *x*.

[*Example*:

`TAN(-1)` results in `-1.557407725`
`TAN(0)` results in `0`
`TAN(1)` results in `1.557407725`

end example]

18.17.7.317 TANH

Syntax:

TANH (*x*)

Description: Computes the hyperbolic tangent of *x*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Arguments:

Name	Type	Description
x	number	The number whose hyperbolic tangent is to be computed.

Return Type and Value: number – The hyperbolic tangent of *x*

[*Example*:

`TANH(-1)` results in `-0.761594156`
`TANH(0)` results in `0`
`TANH(1)` results in `0.761594156`

end example]

18.17.7.318 TBILLEQ

`TBILLEQ ( `*settlement*` , `*maturity*` , `*discount*` )`

Description: Computes the bond-equivalent yield for a U.S. Treasury bill.

Arguments:

Name	Type	Description
settlement	number	The Treasury bill's settlement date. Any time information in the date is ignored.
maturity	number	The Treasury bill's maturity date. Any time information in the date is ignored.
discount	number	The Treasury bill's discount rate.

Return Type and Value: number – The bond-equivalent yield for a U.S. Treasury bill.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* > *maturity*, `#NUM!` is returned.
- *maturity* is more than one year after *settlement*, `#NUM!` is returned.
- *discount* ≤ 0, `#NUM!` is returned.

[*Example*:

`TBILLEQ(DATE(2008,3,31),DATE(2008,6,1),0.0914)` results in `9.4151%`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.319 TBILLPRICE

`TBILLPRICE ( settlement , maturity , discount )`

Description: Computes the price per $100 face value for a U.S. Treasury bill.

Mathematical Formula:

$$TBILLPRCE = 100 \times \left(1 - \frac{discunt \times DSM}{360}\right)$$

where:

- *discount* = argument *discount*
- *DSM* = number of days from settlement to maturity, excluding any maturity date that is more than one calendar year after the settlement date.

Arguments:

Name	Type	Description
settlement	number	The Treasury bill's settlement date. Any time information in the date is ignored.
maturity	number	The Treasury bill's maturity date. Any time information in the date is ignored.
discount	number	The Treasury bill's discount rate.

Return Type and Value: number – The price per $100 face value for a U.S. Treasury bill.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *maturity* is more than one year after *settlement*, `#NUM!` is returned.
- *discount* ≤ 0, `#NUM!` is returned.

[*Example*:

`TBILLPRICE(DATE(2008,3,31),DATE(2008,6,1),0.09)` results in `98.4500`

end example]

18.17.7.320 TBILLYIELD

`TBILLYIELD ( settlement , maturity , pr )`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Computes the yield for a U.S. Treasury bill.

Mathematical Formula:

$$TBILLYIELD = \frac{100 - pr.}{pr.} \times \frac{360}{DSM}$$

where:

- *DSM* = number of days from settlement to maturity, excluding any maturity date that is more than one calendar year after the settlement date.
- *pr.* = argument *pr*

Arguments:

Name	Type	Description
settlement	number	The Treasury bill's settlement date. Any time information in the date is ignored.
maturity	number	The Treasury bill's maturity date. Any time information in the date is ignored.
pr	number	The Treasury bill's price per $100 face value.

Return Type and Value: number – The yield for a U.S. Treasury bill.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *maturity* is more than one year after *settlement*, `#NUM!` is returned.
- *pr* ≤ 0, `#NUM!` is returned.

[*Example*:

`TBILLYIELD(DATE(2008,3,31),DATE(2008,6,1),98.45)` results in `9.1417%`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.17.7.321 TDIST

Syntax:

`TDIST ( x , degrees-freedom , distribution-tails )`

Description: Computes the Percentage Points (probability) for the Student t-distribution where a numeric value, *x*, is a calculated value of *t* for which the Percentage Points are to be computed.

©ISO/IEC 2016 – All rights reserved

Mathematical Formula:

If *distribution-tails* = 1, $TDIST = P(X > x)$, where X is a random variable that follows the t-distribution.

If *distribution-tails* = 2, $TDIST = P(|X| > x) = P(X > x \text{ or } X < -x)$

Arguments:

Name	Type	Description
x	number	The value at which to evaluate the distribution.
degrees-freedom	number	The number of degrees of freedom, truncated to an integer.
distribution-tails	number	The number of distribution tails to return, truncated to an integer. If 1, TDIST returns the one-tailed distribution. If 2, TDIST returns the two-tailed distribution.

Return Type and Value: number – The Percentage Points (probability) for the Student t-distribution.

However, if

- *degrees-freedom* < 1, `#NUM!` is returned.
- *tails* has any value other than 1 or 2, `#NUM!` is returned.
- *x* < 0, `#NUM!` is returned.

[*Example*:

`TDIST(1.959999998,60,1)` results in `0.027322464`
`TDIST(1.959999998,60,2)` results in `0.054644927`

end example]

18.17.7.322 TEXT

Syntax:

`TEXT ( value , format )`

Description: Produces a string containing *value* formatted according to *format*.

Arguments:

Name	Type	Description
value	number	The number that is to be formatted.
format	text	Designates the number, currency, date, or time format to be used. (See §18.8.31 for the set of formats.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: text – The string containing *number* formatted according to *format*.

[*Example*:

`TEXT(1234.567,"$0.00")` results in `$1234.57`
`TEXT(.125,"$0.0%")` results in `12.5%`
`TEXT(1234.567,"YYYY-MM-DD HH:MM:SS")` results in `1903-05-18 13:36:29` in the 1900 date system.

end example]

18.17.7.323 TIME

Syntax:

`TIME ( hour , minute , second )`

Description: Computes the serial date-time for the given time.

Arguments:

Name	Type	Description
hour	number	A number in the range 0–32767, inclusive, truncated to integer, that represents the hour. Any value greater than 23 shall be divided by 24 and the remainder shall be treated as the hour value.
minute	number	A number in the range 0–32767, inclusive, truncated to integer, that represents the minute. Any value greater than 59 shall be converted to the corresponding number of hours and minutes.
second	number	A number in the range 0–32767, inclusive, truncated to integer, that represents the second. Any value greater than 59 shall be converted to the corresponding number of hours, minutes, and seconds.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The serial date-time for the given time, as a value greater than or equal to 0 and less than or equal to 1.

However, if *hour*, *minute*, or *second* are out of range, `#NUM!` is returned.

[*Example*: The following serial date-times are displayed with 16 decimal places.

`TIME(0,0,0)` results in a serial date-time of `0.0000000000000000`
`TIME(0,0,1)` results in a serial date-time of `0.0000115740740741`
`TIME(0,0,2)` results in a serial date-time of `0.0000231481481481`

©ISO/IEC 2016 – All rights reserved

`TIME(0,0,20)` results in a serial date-time of `0.0002314814814815`
`TIME(2,3,20)` results in a serial date-time of `0.0856481481481481`
`TIME(12,0,0)` results in a serial date-time of `0.5000000000000000`
`TIME(23,59,59)` results in a serial date-time of `0.9999884259259260`
`TIME(26,120,240)` results in a serial date-time of `0.1694444444444450`

end example]

18.17.7.324 TIMEVALUE

Syntax:

`TIMEVALUE ( `*date-time-string*` )`

Description: Computes the serial date-time of the time represented by the string *date-time-string*.

Arguments:

Name	Type	Description
date-time-string	text	The date and/or time whose time component serial date-time is to be computed. *date-time-string* can have any date and/or time format. Any date information in *date-time-string* shall be ignored.

Return Type and Value: number – The serial date-time of the time represented by the string *date-time-string*, as a value greater than or equal to 0 and less than or equal to 1.

However, if *date-time-string* is ill-formed, `#VALUE!` is returned.

[*Example*: The following serial date-times are displayed with 16 decimal places.

`TIMEVALUE("10:02:34 ")` results in `0.4184490740740740`
`TIMEVALUE("01-Feb-2006 10:15:29 AM")` results in `0.4274189814823330`
`TIMEVALUE("22:02")` results in `0.9180555555555560`

end example]

18.17.7.325 TINV

Syntax:

`TINV ( `*probability*` , `*degrees-freedom*` )`

Description: Computes the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
probability	number	A probability associated with the two-tailed Student's t-distribution.
degrees-freedom	number	The number of degrees of freedom with which to characterize the distribution, truncated to an integer.

Return Type and Value: number – The t-value of the Student's t-distribution.

However, if

- *probability* < 0 or *probability* > 1, `#NUM!` is returned.
- *degrees-freedom* < 1, `#NUM!` is returned.

[*Example*:

`TINV(0.054644927,60)` results in `1.95999999`

end example]

18.17.7.326 TODAY

Syntax:

```
TODAY ( )
```

Description: Computes the serial date-time of the current date, taking into account the current date base system.

Arguments: None.

Return Type and Value: number – The serial date-time of the current date.

[*Example*:

On February 25, 2006, `TODAY()` results in 38773 for the 1900 date system, or 37311 for the 1904 date system

end example]

18.17.7.327 TRANSPOSE

Syntax:

```
TRANSPOSE ( array )
```

Description: Creates a new array that is the transpose of an existing array, by copying the first row of the existing array to the first column of the new array, the second row of the existing array as the second column of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

the new array, and so on. The formula containing the call to TRANSPOSE shall be an array formula in a range that has the same number of rows and columns, respectively, as *array* has columns and rows.

Arguments:

Name	Type	Description
array	array, reference	The set of values to be transposed.

Return Type and Value: array – The new array.

[*Example*:

`TRANSPOSE({10,20,30})` results in the array `{10;20;30}`

end example]

18.17.7.328 TREND

Syntax:

TREND (*known-ys* [, [*known-xs*] [, [*new-xs*] [, *const-flag*]]])

Description: Computes values along a linear trend. Fits a straight line (using the method of least squares) to the arrays *known-ys* and *known-xs*. The y-values along that line for the array of *new-xs* specified.

Arguments:

Name	Type	Description
known-ys	array	The set of y-values already known in the relationship *y=mx+b*. If that array is in a single column, each column of *known-xs* is interpreted as a separate variable. If that array is in a single row, each row of *known-xs* is interpreted as a separate variable.
known-xs	array	An optional set of x-values that might already be known in the relationship *y=mx+b*. The array *known-xs* can include one or more sets of variables. If only one variable is used, *known-ys* and *known-xs* can be ranges of any shape, as long as they have equal dimensions. If more than one variable is used, *known-ys* shall be a vector. If *known-xs* is omitted, it is assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
new-xs	array	New x-values for which TREND is to return corresponding y-values. *new-xs* shall include a column (or row) for each independent variable, just as *known-xs* does. So, if *known-ys* is in a single column, *known-xs* and *new-xs* shall have the same number of columns. If *known-ys* is in a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		single row, *known-xs* and *new-xs* shall have the same number of rows. If *new-xs* is omitted, it is assumed to be the same as *known-xs*. If both *known-xs* and *new-xs* are omitted, they are assumed to be the array {1,2,3,…} that is the same size as *known-ys*.
const-flag	logical	Specifies whether to force the constant b to equal 0. If TRUE or omitted, *b* is calculated normally. If FALSE, *b* is set equal to 0 and the m-values are adjusted so that *y=mx*.

Return Type and Value: array – The values along a linear trend, as an array of numbers.

18.17.7.329 TRIM

Syntax:

```
TRIM ( string )
```

Description: Makes a string that is a copy of *string* with the leading and trailing space characters removed, and each sequence of embedded spaces reduced to a single space. The space character referred to here is character U+0020.

Arguments:

Name	Type	Description
string	text	Designates the string to be trimmed.

Return Type and Value: text – The trimmed copy of *string*.

[*Example*:

`TRIM("  abc  def  ")` results in `abc def`

end example]

18.17.7.330 TRIMMEAN

Syntax:

```
TRIMMEAN ( array , percent )
```

Description: Computes the mean of the interior of a data set by excluding a percentage of data points from the top and bottom tails of a data set. `TRIMMEAN` rounds the number of excluded data points down to the nearest multiple of 2. For symmetry, `TRIMMEAN` excludes the same number of data points from the top and bottom of the data set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Arguments:

Name	Type	Description
array	array, reference	The numeric values to trim and average.
percent	number	The fractional number of data points to exclude from the calculation. [*Example*: If *percent* = 0.2, 4 points are trimmed from a data set of 20 points (20x0.2): 2 from the top and 2 from the bottom of the set. *end example*]

Return Type and Value: number – The mean of the interior of a data set.

However, if *percent* < 0 or *percent* ≥ 1, `#NUM!` is returned.

[*Example*:

`TRIMMEAN({4,6,2,5,7,8,9},0.2)` results in `5.857142857`

end example]

18.17.7.331 TRUE

Syntax:

```
TRUE ( )
```

Description: Computes the value TRUE. (A call to function TRUE is equivalent to using the *logical-constant* TRUE.)

Arguments: None.

Return Type and Value: logical – The value TRUE.

[*Example*:

`TRUE()` results in TRUE

end example]

18.17.7.332 TRUNC

Syntax:

```
TRUNC ( x [ , number-digits ] )
```

Description: Truncates *x* to the number of fractional digits by *number-digits*.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
x	array, reference	The value to be rounded down.
number-digits	number	The number of fractional digits to which *x* is to be truncated. The default value for *number-digits* is 0.

Return Type and Value: number – The truncated value of *x*.

[*Example*:

`TRUNC(PI())` results in `3`
`TRUNC(PI(),1)` results in `3.1`
`TRUNC(PI(),3)` results in `3.141`
`TRUNC(PI(),5)` results in `3.14159`

end example]

18.17.7.333 TTEST

Syntax:

`TTEST` (*array-1* , *array-2* , *distribution-tails* , *test-type*)

Description: Computes the probability associated with a Student's t-Test.

Arguments:

Name	Type	Description
array-1	array, reference	The first numerical data set.
array-1	array, reference	The first numerical data set.
distribution-tails	number	Specifies the number of distribution tails, truncated to an integer. If 1, TTEST uses the one-tailed distribution. If 2, TTEST uses the two-tailed distribution.
test-type	number	The truncated-to-integer kind of t-Test to perform, as follows:

Value	Test Performed
1	Paired
2	Two-sample equal variance (homoscedastic)
3	Two-sample unequal variance (heteroscedastic)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The probability associated with a Student's t-Test.

However, if

- *array-1* and *array-2* have a different number of data points, and *test-type* is 1, the return value is unspecified.
- *distribution-tails* is any value other than 1 or 2, `#NUM!` is returned.
- *test-type* is any value other than 1, 2 or 3, `#NUM!` is returned.

[*Example*: Given the following data:

	A	B
1	Data 1	Data 2
2	3	6
3	4	19
4	5	3
5	8	2
6	9	14
7	1	4
8	2	5
9	4	17
10	5	1

`TTEST(A2:A10,B2:B10,2,1)` results in `0.196016`

end example]

18.17.7.334 TYPE

Syntax:

TYPE (*value*)

Description: Computes the type of *value* or, if *value* is a reference to a single cell, the type of the value in that cell.

Arguments:

Name	Type	Description
value	any	The value whose type is to be determined. No conversion shall take place on an argument passed to this function.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – An integer that indicates the type of *value* or, if *value* is a reference to a single cell, the type of the value in that cell, as follows:

Type of *value*	Value Returned
number	1
text	2
logical	4
error value	16
array of any kind	64

[*Example*:

`TYPE(10.5)` results in `1`
`TYPE(A10)` results in `1`, when A10 contains a number
`TYPE("ABC")` results in `2`
`TYPE(A10)` results in `2`, when A10 contains a string
`TYPE(TRUE)` results in `4`
`TYPE(A10)` results in `4`, when A10 contains a logical value

`TYPE(5/0)` results in `16`
`TYPE(A10)` results in `16`, when A10 contains any error value
`TYPE({1,2,3})` results in `64`
`TYPE({TRUE,2.5,#N/A})` results in `64`
`TYPE(IF(10>5,"Yes",20))` results in `2`
`TYPE(IF(10<5,"Yes",20))` results in `1`

end example]

18.17.7.335 UPPER

Syntax:

```
UPPER ( string )
```

Description: Makes an uppercase version of *string*.

Arguments:

Name	Type	Description
string	text	Designates the string to be converted.

Return Type and Value: text – The uppercase version of *string*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*:

UPPER("AbCd123#$%^") results in ABCD123#$%^
UPPER(A10) results in 234FRTQWC$#%, when A10 contains 234FRTqwc$#%

end example]

18.17.7.336 USDOLLAR

Syntax:

USDOLLAR (*number* [, *num-decimal*])

Description: Produces a string containing *number* rounded to *num-decimal* decimal places. The thousands separator is the comma, the radix point is the period, and the currency symbol is "$". The format used is $#,##0.00;($#,##0.00).

Arguments:

Name	Type	Description
number	number	The number that is to be formatted.
num-decimal	number	Designate the number of decimal places to be used in the resulting string; it is truncated to an integer. If *num-decimal* is negative, *number* is rounded to the left of the decimal point. If omitted, a value of 2 shall be assumed.

Return Type and Value: text – The string containing *number* rounded to *num-decimal* decimal places, and have a currency symbol and thousands separators.

[*Example*:

USDOLLAR(1234.567) results in $1,234.57
USDOLLAR(1234.567,-2) results in $1,200
USDOLLAR(-1234.567,4) results in ($1,234.5670)

end example]

18.17.7.337 VALUE

Syntax:

VALUE (*string*)

Description: Converts *string* to a number.

Arguments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
string	text	Designates a string that contains a number formatted using any number, currency, date, or time format. (See §18.8.31 for the set of formats.) Date and time strings are converted to their equivalent serial date-time.

Return Type and Value: number – The number represented by *string*.

[*Example*:

`VALUE("123.456")` results in `123.456`
`VALUE("$1,000")` results in `1000`
`VALUE("23-Mar-2002")` results in the corresponding serial date-time
`VALUE("16:48:00")-VALUE("12:17:12")` results in `0.188056`

end example]

18.17.7.338 VAR

Syntax:

VAR (*argument-list*)

Description: Makes an estimate of the variance based on a sample. [*Note*: VAR assumes that its arguments are a sample of the population. If the data represents the entire population, VARP should be used instead. If logical values and text representations of numbers in a reference are to be included as part of the calculation, use VARA instead. *end note*]

Mathematical Formula:

$$\frac{\sum(x-\bar{x})^2}{(n-1)}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-1*,…, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *argument*s in *argument-list* designate the numbers that are samples of the population. Logical values, and text representations of numbers that are entered directly

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Name	Type	Description
		into the list of arguments are included. If an argument is an array or reference, only numbers in that array or reference are included. Empty cells, logical values, text, or error values in the array or reference are ignored.

Return Type and Value: number – An estimate of the variance based on a sample.

However, if the sample size ≤ *1*, `#DIV/0!` is returned.

[*Example*:

`VAR(1202,1220,1323,1254,1302)` results in `2683.2`

end example]

18.17.7.339 VARA

Syntax:

VARA (*argument-list*)

Description: Makes an estimate of the variance based on a sample. [*Note*: VARA assumes that its arguments are a sample of the population. If the data represents the entire population, VARPA should be used instead. If logical values and text representations of numbers in a reference are to be excluded as part of the calculation, use VAR instead. *end note*]

Mathematical Formula:

$$\frac{\sum(x-\bar{x})^2}{(n-1)}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-1*,…, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *argument*s in *argument-list* designate the numbers that are samples of the population. Logical values and text representations of numbers that are entered directly into the list of arguments are included. Arguments that contain TRUE evaluate as 1; arguments that contain text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		or FALSE evaluate as zero. If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored.

Return Type and Value: number – An estimate of the variance based on a sample.

However, if the sample size ≤ *1*, `#DIV/0!` is returned.

[*Example*:

`VARA(1202,1220,1323,1254,1302)` results in `2683.2`

end example]

18.17.7.340 VARP

Syntax:

`VARP` (*argument-list*)

Description: Computes the variance of an entire population. [*Note*: VARP assumes that its arguments are the total population. If the data represents a population sample only, VAR should be used instead. If logical values and text representations of numbers in a reference are to be included as part of the calculation, use VARPA instead. *end note*]

Mathematical Formula:

$$\frac{\sum(-\bar{x})^2}{n}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean `AVERAGE(`*argument-1*`,` *argument-1*`,…,` *argument-n*`)`

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *argument*s in *argument-list* designate the numbers that are the members of the population. Logical values, and text representations of numbers that are entered directly into the list of arguments are included. If an argument is an array or reference, only numbers in that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		array or reference are included. Empty cells, logical values, text, or error values in the array or reference are ignored.

Return Type and Value: number – The variance of an entire population.

However, if the sample size is zero, `#DIV/0!` is returned.

[*Example*:

`VARP(1202,1220,1323,1254,1302)` results in `2146.56`

end example]

18.17.7.341 VARPA

Syntax:

VARPA (*argument-list*)

Description: Makes the variance of an entire population. [*Note:* VARPA assumes that its arguments are the total population. If the data represents a population sample only, VARA should be used instead. If logical values and text representations of numbers in a reference are to be excluded as part of the calculation, use VARP instead. *end note*]

Mathematical Formula:

$$\frac{\sum(x-\bar{x})^2}{n}$$

where:

- n = the sample size
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*argument-1*, *argument-1*,…, *argument-n*)

Arguments:

Name	Type	Description
argument-list	logical, number, name, text, array, reference	The *argument*s in *argument-list* designate the numbers that are the members of the population.

Return Type and Value: number – The variance of an entire population.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Arguments can be numbers; names, arrays, or references that contain numbers; text representations of numbers; or logical values, in a reference. Text representations of numbers that are entered directly into the list of arguments are included. Arguments that contain TRUE evaluate as 1; arguments that contain text or FALSE evaluate as zero. If an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored.

However, if the sample size is zero, `#DIV/0!` is returned.

[*Example*:

`VARPA(1202,1220,1323,1254,1302)` results in `2146.56`

end example]

18.17.7.342 VDB

VDB (*cost* , *salvage* , *life* , *start-period* , *end-period* [, [[*factor*] [, [*no-switch-flag*]]]])

Description: Computes the depreciation of an asset for the period specified, including partial periods, using the double-declining balance or some other specified method.

Arguments:

Name	Type	Description
cost	number	The number *cost* is the initial cost of the asset.
salvage	number	The value at the end of the depreciation. (This is sometimes called the salvage value of the asset.) This value can be 0.
life	number	The number of periods over which the asset is being depreciated. (This is sometimes called the useful life of the asset.)
start-period	number	The starting period for which the depreciation is to be calculated. (*start-period* shall use the same units as *life*.)
end-period	number	The ending period for which the depreciation is to be calculated. (*end-period* shall use the same units as *life*.)
factor	number	The rate at which the balance declines. If omitted, it is assumed to be 2 (the double-declining balance method).
no-switch-flag	logical	Specifies whether to switch to straight-line depreciation when depreciation is greater than the declining balance calculation. If TRUE, straight-line depreciation is not used even when the depreciation is greater than the declining balance calculation. If FALSE or omitted, the straight-line depreciation is used when depreciation is greater than the declining balance calculation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Return Type and Value: number – The depreciation of an asset for the period specified.

However, if:

- any numerical argument value is non-positive, `#NUM!` is returned.
- *cost* is greater than *salvage*, `zero` is returned.
- *start-period* is greater than *end-period*, `#NUM!` is returned.
- *end-period* is greater than *life*, `#NUM!` is returned.

[*Example*:

`VDB(2400,300,10*365,0,1)` results in `1.32`
`VDB(2400,300,10*12,0,1)` results in `40.00`
`VDB(2400,300,10*12,6,18)` results in `396.31`

end example]

18.17.7.343 VLOOKUP

Syntax:

VLOOKUP (*lookup-value* , *table-array* , *col-index-num* [, [*range-lookup-flag*]])

Description: Performs a vertical search for a value in the left-most column of a table or an array, noting the row in which the matching value is found. From that row, the value from a given column is returned.

Arguments:

Name	Type	Description
lookup-value	value of any type or a reference to a value of any type.	The value to be located in the left-most column of the table. If *range-lookup* is FALSE and *lookup-value* is a string, the wildcard characters, question mark (?) and asterisk (*), can be included in *lookup-value*. A question mark matches any single character; an asterisk matches any sequence of characters. To find a question mark or asterisk, use a tilde (~) before the character.
table-array	array, reference, name	Designates the table of information to be searched. The values in the left-most column of *table-array* can be text, numbers, or logical values. The values in the left-most column of *table-array* shall be placed in "ascending order", as follows: …, -2, -1, 0, 1, 2, …, A–Z, FALSE, TRUE. Uppercase and lowercase text is treated as equivalent.
col-index-num	number	The column number in *table-array* from which the matching value is to be returned. (A *col-index-num* of 1 returns the left-most column value in *table-array*, a *col-*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		index-num of 2 returns the next column in *table-array*, and so on.)
range-lookup-flag	logical	Specifies whether HLOOKUP is to find an exact or approximate match. If TRUE or omitted, an approximate match is returned. That is, if an exact match is not found, the next largest value that is less than *lookup-value* is returned. If FALSE, an exact match is performed, in which case, the values in the left-most column of *table-array* need not be sorted. If there are two or more values in the left-most column of *table-array* that match *lookup-value*, the top-most value found is used.

Return Type and Value: any – The value from a given row number, where the column is determined by a search of the top row looking for a match with a given value.

However, if

- An exact match is performed, but no match is found, #N/A is returned.
- *col-index-num* is less than 1, #VALUE! is returned.
- *col-index-num* is greater than the number of columns in *table-array*, #REF! is returned.
- *lookup-value* is smaller than the smallest value in the left-most column of *table-array*, #N/A is returned.

[*Example*: Given the following data:

	A	B	C
1	Density	Bearings	Bolts
2	0.457	3.55	500
3	0.525	3.25	400
4	0.616	2.93	300
5	0.675	2.75	250
6	0.746	2.57	200
7	0.835	2.38	150
8	0.946	2.17	100
9	1.09	1.95	50
10	1.29	1.71	0

`VLOOKUP(1,A2:C10,2)` results in `2.17`
`VLOOKUP(1,A2:C10,3,TRUE)` results in `100.00`
`VLOOKUP(2,A2:C10,2,TRUE)` results in `1.71`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

18.17.7.344 WEEKDAY

Syntax:

WEEKDAY (*serial-value* [, *weekday-start-flag*])

Description: Computes the weekday number for the date having the given *serial-value*, taking into account the current date system and *weekday-start-flag*, if present. See §18.17.4.1 for special handling of certain days in 1900.

Arguments:

Name	Type	Description
serial-value	number	The date whose weekday number is to be computed. The value of *serial-value* is truncated to an integer.
weekday-start-flag	number	When truncated to integer, indicates the weekday numbering convention to be used, as follows:

Value	Meaning
1 or omitted	1 (Sunday) through 7 (Saturday)
2	1 (Monday) through 7 (Sunday)
3	0 (Monday) through 6 (Sunday)
11	1 (Monday) through 7 (Sunday)
12	1 (Tuesday) through 7 (Monday)
13	1 (Wednesday) through 7 (Tuesday)
14	1 (Thursday) through 7 (Wednesday)
15	1 (Friday) through 7 (Thursday)
16	1 (Saturday) through 7 (Friday)
17	1 (Sunday) through 7 (Saturday)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Return Type and Value: number – The weekday number for the date having the given serial date-time.

However, if

- *serial-value* is out of range for the current date system, #NUM! is returned.
- *weekday-start-flag* is out of the range specified in the table above, #NUM! is returned.

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

`WEEKDAY(DATE(2006,2,1))` results in 4 (Wednesday)
`WEEKDAY(DATE(2006,2,1),11)` results in 3 (Wednesday)
`WEEKDAY(DATE(2006,2,1),12)` results in 2 (Wednesday)
`WEEKDAY(DATE(2006,2,1),3)` results in 2 (Wednesday)

end example]

18.17.7.345 WEEKNUM

Syntax:

`WEEKNUM ( serial-value [ , weekday-start-flag ] )`

Description: Computes the week number of the date corresponding to *serial-value*. The function allows two number systems:

- System 1: The week containing January 1 is the first week of the year, and is numbered week 1.
- System 2: The week containing the first Thursday of the year is the first week of the year, and is numbered as week 1 [ISO 8601].

Arguments:

Name	Type	Description
serial-value	number	The date whose week number is to be computed. The value of *serial-value* is truncated to an integer.
weekday-start-flag	number	When truncated to integer, indicates the weekday on which the week begins, as follows:

weekday-start-flag	Meaning	Number System
1 or omitted	Week begins on Sunday.	System 1
2	Week begins on Monday.	System 1
11	Week begins on Monday.	System 1
12	Week begins on Tuesday.	System 1
13	Week begins on Wednesday.	System 1
14	Week begins on Thursday.	System 1

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		15 | Week begins on Friday. | System 1 16 | Week begins on Saturday. | System 1 17 | Week begins on Sunday. | System 1 21 | Week begins on Monday. | System 2

Return Type and Value: number – The week number of the date corresponding to *serial-value*.

However, if

- *serial-value* is out of range for the current date system, `#NUM!` is returned.
- *weekday-start-flag* is out of the range specified in the table above, `#NUM!` is returned.

[*Example*:

`WEEKNUM(DATE(2006,1,1)` results in `1`
`WEEKNUM(DATE(2006,1,1),1)` results in `1`
`WEEKNUM(DATE(2006,1,1),17)` results in `1`
`WEEKNUM(DATE(2006,1,1),21)` results in `1`
`WEEKNUM(DATE(2006,2,1),1)` results in `5`
`WEEKNUM(DATE(2006,2,1),2)` results in `6`
`WEEKNUM(DATE(2006,2,1),11)` results in `6`

end example]

18.17.7.346 WEIBULL

Syntax:

`WEIBULL` (*x* , *alpha* , *beta* , *cumulative-flag*)

Description: Computes the Weibull distribution.

Mathematical Formula:

The equation for the Weibull cumulative distribution function is:

$$F(x, \alpha, \beta) = 1 - e^{-(x/\beta)^{\alpha}}$$

The equation for the Weibull probability density function is:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

$$f(x,\alpha,\beta) = \frac{}{\beta^{\alpha}} x^{\alpha-1} e^{-(x/\beta)^{\alpha}}$$

When *alpha* = 1, `WEIBULL` returns the exponential distribution with:

$$\lambda = \frac{1}{\beta}$$

where:

- *x* = argument *x*
- α = argument *alpha*
- β = argument *beta*

Arguments:

Name	Type	Description
x	number	The value at which the distribution is to be evaluated.
alpha	number	A parameter of the distribution.
beta	number	A parameter of the distribution.
cumulative-flag	logical	Determines the form of the function. If `TRUE`, `WEIBULL` returns the cumulative distribution function; if `FALSE`, it returns the probability density function.

Return Type and Value: number – The Weibull distribution.

However, if

- *x* < 0, `#NUM!` is returned.
- *alpha* ≤ 0, `#NUM!` is returned.
- *beta* ≤ 0, `#NUM!` is returned.

[*Example*:

`WEIBULL(105,20,100,TRUE)` results in `0.92958139`
`WEIBULL(105,20,100,FALSE)` results in `0.035588864`

end example]

18.17.7.347 WORKDAY

Syntax:

`WORKDAY ( start-date , day-offset [ , holidays ] )`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Description: Computes the serial date-time of the date that is *day-offset* working days offset from *start-date*. Weekend days (Saturday and Sunday) and any holidays specified by *holidays* are not considered as working days.

Arguments:

Name	Type	Description
start-date	number	The start date, truncated to integer.
day-offset	number	The number of working days before or after *start-date*. A positive value yields a future date; a negative value yields a past date; a zero value yields the date *start-date*. *day-offset* is truncated to an integer.
holidays	reference, array	An optional set of one or more dates that are to be excluded from the working day calendar. *holidays* shall be a range of cells that contain the dates, or an array constant of the serial date-times that represent those dates. The ordering of dates or serial date-times in *holidays* can be arbitrary.

Return Type and Value: number – The serial date-time of the date that is *day-offset* working days offset from *start-date*, excluding the specified holidays.

However, if

- *start-date* is out of range for the current date system, `#NUM!` is returned.
- Any date in *holidays* is out of range for the current date system, `#NUM!` is returned.
- *start-date* plus *day-offset* does not yield a date, `#NUM!` is returned.

[*Example*:

`WORKDAY(DATE(2006,1,1),0)` results in a serial date-time corresponding to 1-Jan-2006
`WORKDAY(DATE(2006,1,1),10)` results in a serial date-time corresponding to 13-Jan-2006
`WORKDAY(DATE(2006,1,1),-10)` results in a serial date-time corresponding to 19-Dec-2005
`WORKDAY(DATE(2006,1,1),20,{"2006/1/2","2006/1/16"})` results in a serial date-time corresponding to 31-Jan-2006

end example]

18.17.7.348 WORKDAY.INTL

Syntax:

Number form: `WORKDAY.INTL` (*start-date* , *day-offset* [,[*weekend-number*] [, *holidays*]])

String form: `WORKDAY.INTL` (*start-date* , *day-offset* [, [*weekend-string*] [, *holidays*]])

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Description: Computes the serial date-time of the date that is *day-offset* working days offset from *start-date*. Weekend days and any holidays specified by *holidays* are not considered as working days.

Arguments:

Name	Type	Description
start-date	number	The start date, truncated to integer.
day-offset	number	The number of working days before or after *start-date*. A positive value yields a future date; a negative value yields a past date; a zero value yields the date *start-date*. *day-offset* is truncated to an integer.
weekend-number	number	Indicates the days of the week that are weekend days and are not considered working days. Values are shown in the table below
weekend-string	string	Indicates the days of the week that are weekend days and are not considered working days. Values of *weekend-string* are seven characters long and each character in the string represents a day of the week, beginning with Monday. [*Example*: "0000011" would result in a weekend that is Saturday and Sunday. *end example*]
holidays	reference, array	An optional set of one or more dates that are to be excluded from the working day calendar. *holidays* shall be a range of cells that contain the dates, or an array constant of the serial date-times that represent those dates. The ordering of dates or serial date-times in *holidays* can be arbitrary.

weekend-number	Weekend days
1 or omitted	Saturday, Sunday
2	Sunday, Monday
3	Monday, Tuesday
4	Tuesday, Wednesday
5	Wednesday, Thursday
6	Thursday, Friday
7	Friday, Saturday
11	Sunday only
12	Monday only
13	Tuesday only

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

weekend-number	Weekend days
14	Wednesday only
15	Thursday only
16	Friday only
17	Saturday only

Return Type and Value: number – The serial date-time of the date that is *day-offset* working days offset from *start-date*, excluding the specified weekend days and holidays.

However, if

- *start-date* is out of range for the current date system, `#NUM!` is returned.
- Any date in *holidays* is out of range for the current date system, `#NUM!` is returned.
- *start-date* plus *day-offset* does not yield a date, `#NUM!` is returned.

[*Example*:

`WORKDAY.INTL(DATE(2006,1,1),0)` results in a serial date-time corresponding to 1-Jan-2006
`WORKDAY.INTL(DATE(2006,1,1),10)` results in a serial date-time corresponding to 13-Jan-2006
`WORKDAY.INTL(DATE(2006,1,1),10,7)` results in a serial date-time corresponding to 13-Jan-2006
`WORKDAY.INTL(DATE(2006,1,1),-10)` results in a serial date-time corresponding to 19-Dec-2005
`WORKDAY.INTL(DATE(2006,1,1),20,1,{"2006/1/2","2006/1/16"})` results in a serial date-time corresponding to 31-Jan-2006
`WORKDAY.INTL(DATE(2006,1,1),20,"0000011",{"2006/1/2","2006/1/16"})` results in a serial date-time corresponding to 31-Jan-2006

end example]

18.17.7.349 XIRR

`XIRR ( values , dates [ , [ guess ] ] )`

Description: Computes the internal rate of return for a schedule of cash flows that is not necessarily periodic. `XIRR` uses an iterative calculation technique that cycles through the calculation until the result is accurate within 0.000001 percent.

Mathematical Formula:

Using a changing rate (starting with *guess*), `XIRR` cycles through the calculation until the result is accurate within 0.000001 percent. The rate is changed until:

$$0 = \sum_{i=1}^{N} \frac{P_i}{(1 + rate)^{\frac{(d_i - d_1)}{365}}}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

where:

- d_i = the i^{th}, or last, payment date.
- d_1 = the 0^{th} payment date.
- P_i = the i^{th}, or last, payment.

Arguments:

Name	Type	Description
values	array, reference	A series of cash flows that corresponds to a schedule of payment dates specified in *dates*. The first payment is optional and corresponds to a cost or payment that occurs at the beginning of the investment. If the first value is a cost or payment, it shall have a negative value. All succeeding payments are discounted based on a 365-day year. The series of values shall contain at least one positive and one negative value.
dates	reference	A schedule of payment dates that corresponds to the cash flow payments in *values*. The first payment date indicates the beginning of the schedule of payments. All other dates shall be later than this date, but they can occur in any order. Time information in the date arguments is ignored.
guess	number	An estimate of the result of `XIRR`. If omitted, it is assumed to be 0.1 (i.e., 10 percent).

Return Type and Value: number – The internal rate of return for a schedule of cash flows that is not necessarily periodic.

However, if

- Any date in *dates* is out of range for the current date system, `#NUM!` is returned.
- *values* does not contain at least one positive and one negative value, `#NUM!` is returned.
- Any date in *dates* precedes the starting date, `#NUM!` is returned.
- *values* and *dates* contain different numbers of values, `#NUM!` is returned.
- The calculation has not converged after an implementation-defined number of tries, `#NUM!` is returned.

[*Example*: When the cells `F2397:J2397`contain the dates January 1, 2008; March 1,2008; October 30, 2008; February 15, 2009, and April 1, 2009:

`XIRR({-10000,2750,4250,3250,2750},F2397:J2397,0.1)` results in `37.34%`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.17.7.350 XNPV

XNPV (*rate* , *values* , *dates*)

Description: Computes the net present value for a schedule of cash flows that is not necessarily periodic.

Mathematical Formula:

$$XNPV = \sum_{i=1}^{N} \frac{P_i}{(1 + rte)^{\frac{(d_i - d_1)}{365}}}$$

where:

- d_i = the i^{th}, or last, payment date.
- d_1 = the 0^{th} payment date.
- P_i = the i^{th}, or last, payment.

Arguments:

Name	Type	Description
rate	number	The discount rate to apply to the cash flows.
values	array, reference	A series of cash flows that corresponds to a schedule of payment dates specified in *dates*. The first payment is optional and corresponds to a cost or payment that occurs at the beginning of the investment. If the first value is a cost or payment, it shall have a negative value. All succeeding payments are discounted based on a 365-day year. The series of values shall contain at least one positive and one negative value.
dates	reference	A schedule of payment dates that corresponds to the cash flow payments in *values*. The first payment date indicates the beginning of the schedule of payments. All other dates shall be later than this date, but they can occur in any order. Time information in the date arguments is ignored.

Return Type and Value: number – The net present value for a schedule of cash flows that is not necessarily periodic.

However, if

- *rate* ≤ 0, `#NUM!` is returned.
- Any date in *dates* is out of range for the current date system, `#NUM!` is returned.
- Any date in *dates* precedes the starting date, `#NUM!` is returned.
- *values* and *dates* contain different numbers of values, `#NUM!` is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: When the cells F2397:J2397 contain the dates January 1, 2008; March 1,2008; October 30, 2008; February 15, 2009, and April 1, 2009:

XNPV(0.09,{-10000,2750,4250,3250,2750},F2397:J2397) results in 2086.65

end example]

18.17.7.351 YEAR

Syntax:

YEAR (*date-value*)

Description: Computes the numeric year in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*, taking into account the current date system. That date and/or time shall be expressed either as a serial date-time, in which case, its fractional part is ignored, or as a *string-constant* having any date and/or time format, in which case, any time information shall be ignored.

Arguments:

Name	Type	Description
date-value	number, text	The date and/or time whose year is to be computed. That date and/or time shall be expressed either as a serial date-time, in which case, its fractional part is ignored, or as a *string-constant* having any date and/or time format, in which case, any time information shall be ignored.

Return Type and Value: number – The year in the Gregorian calendar [ISO 8601 §3.2.1] for the date and/or time having the given *date-value*. The range of return values is determined by the date system currently in use (§18.17.4).

However, if *date-value* is out of range for the current date system, #NUM! is returned.

[*Example*:

YEAR(DATE(2006,1,2)) results in 2006
YEAR(DATE(2006,0,2)) results in 2005
YEAR("2006/1/2 10:45 AM") results in 2006
YEAR(30000) results in 1982 for the 1900 date system, or 1986 for the 1904 date system

end example]

18.17.7.352 YEARFRAC

Syntax:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

YEARFRAC (*start-date* , *end-date* [, *basis*])

Description: Computes the fractional number of years represented by the number of whole days between two dates, *start-date* and *end-date*., according to *basis*.

Arguments:

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>start-date</td><td>number</td><td>The period's starting date. start-date can be earlier than, the same as, or later than end-date.</td></tr>
<tr><td>end-date</td><td>number</td><td>The period's ending date.</td></tr>
<tr><td>basis</td><td>number</td><td>The truncated integer type of day count basis to use, as follows:
<table>
<tr><th>Value</th><th>Day Count Basis</th></tr>
<tr><td>0 or omitted</td><td>US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:
If the date is 28 or 29 February, it is adjusted to 30 February.
For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.</td></tr>
<tr><td>1</td><td>Actual/actual. The actual number of days</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
			between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days. Year length used is the average length of the years that the range crosses (regardless of where the start and end dates fall in their respective years).
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		situations where the first date is 28 or 29 Februar y.

All arguments are truncated to integers.

Return Type and Value: number – The fractional number of years represented by the number of whole days between two dates, *start-date* and *end-date*., according to *basis*. If the Actual/actual basis is used, the year length used is the average length of the years that the range crosses, regardless of where *start-date* and *end-date* fall in their respective years.

However, if the value of *basis* is out of range, `#NUM!` is returned.

[*Example*:

`YEARFRAC(DATE(2006,1,1),DATE(2006,3,26))` results in `0.236111111`
`YEARFRAC(DATE(2006,3,26),DATE(2006,1,1))` results in `0.236111111`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1))` results in `0.5`
`YEARFRAC(DATE(2006,1,1),DATE(2007,9,1))` results in `1.666666667`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1),0)` results in `0.5`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1),1)` results in `0.495890411`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1),2)` results in `0.502777778`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1),3)` results in `0.495890411`
`YEARFRAC(DATE(2006,1,1),DATE(2006,7,1),4)` results in `0.5`
`YEARFRAC(DATE(2004,3,1),DATE(2006,3,1),1)` results in 1.998175 (because 2004 is a leap year and Actual/actual basis is used, the average year length is 365.3333)

end example]

18.17.7.353 YIELD

Syntax:

`YIELD` (*settlement* , *maturity* , *rate* , *pr* , *redemption* , *frequency* [, [*basis*]])

Description: Computes the yield on a security that pays periodic interest.

Mathematical Formula:

If there is one coupon period or less until redemption, `YIELD` is calculated as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

$$YIELD = \frac{\left(\frac{redempion}{100} + \frac{rate}{frequency}\right) - \left(\frac{par}{100} + \left(\frac{A}{E} \times \frac{rate}{frequency}\right)\right)}{\frac{par}{100} + \left(\frac{A}{E} \times \frac{rate}{frequency}\right)} \times \frac{frequency \times E}{DSR}$$

where:

- *A* = number of days from the beginning of the coupon period to the settlement date (accrued days).
- *DSR* = number of days from the settlement date to the redemption date.
- *E* = number of days in the coupon period.
- *frequency* = argument *frequency*
- *par* = argument *pr*
- *rate* = argument *rate*
- *redemption* = argument *redemption*

If there is more than one coupon period until redemption, YIELD is calculated through some number of iterations. The resolution uses the Newton method, based on the formula used for the function PRICE. The yield is changed until the estimated price given the yield is close to price.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
rate	number	The security's interest rate.
pr	number	The security's price per 100 currency units face value.
redemption	number	The security's redemption value per 100 currency units face value.
frequency	number	the number of coupon payments per year. (For annual payments, *frequency* is 1; for semiannual payments, *frequency* is 2; for quarterly payments, *frequency* is 4.) *frequency* is truncated to an integer.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
			• If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
			Actual/365. Similar to Basis 1, but always has 365 days per year.
		4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 Februar y.

Time information in the date arguments is ignored.

Return Type and Value: number – The yield on a security that pays periodic interest.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *rate* < 0, `#NUM!` is returned.
- *pr* or *redemption* ≤ 0, `#NUM!` is returned.
- *frequency* is any number other than 1, 2, or 4, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`YIELD(DATE(2008,2,15),DATE(2016,11,15),0.0575,95.04287,100,2,0)` results in `6.5000%`

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

18.17.7.354 YIELDDISC

Syntax:

`YIELDDISC ( `*settlement*` , `*maturity*` , `*pr*` , `*redemption*` [ , [ `*basis*` ] ] )`

Description: Computes the annual yield for a discounted security.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
pr	number	The security's price per 100 currency units face value.
redemption	number	The security's redemption value per 100 currency units face value.
basis	number	The truncated integer type of day count basis to use, as follows:

Value	Day Count Basis
`0` or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		which case it does not change.
		1 — Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2 — Actual/360. Similar to Basis 1, but only has 360 days per year.
		3 — Actual/365. Similar to Basis 1, but always has 365 days per year.
		4 — European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description
		<table><tr><td></td><td>first date is 28 or 29 Februar y.</td></tr></table>

Time information in the date arguments is ignored.

Return Type and Value: number – The annual yield for a discounted security.

However, if

- *settlement* or *maturity* is out of range for the current date system, `#NUM!` is returned.
- *settlement* ≥ *maturity*, `#NUM!` is returned.
- *pr* or *redemption* ≤ 0, `#NUM!` is returned.
- *basis* < 0 or *basis* > 4, `#NUM!` is returned.

[*Example*:

`YIELDDISC(DATE(2008,2,16),DATE(2008,3,1),99.795,100,2)` results in `5.2823%`

end example]

18.17.7.355 YIELDMAT

Syntax:

`YIELDMAT ( `*settlement*` , `*maturity*` , `*issue*` , `*rate*` , `*pr*` [ , [ `*basis*` ] ] )`

Description: Computes the annual yield of a security that pays interest at maturity.

Arguments:

Name	Type	Description
settlement	number	The security's settlement date.
maturity	number	The security's maturity date.
issue	number	The security's issue date.
rate	number	The security's interest rate.
pr	number	The security's price per 100 currency units face value.
basis	number	The truncated integer type of day count basis to use, as follows: <table><tr><th>Value</th><th>Day Count Basis</th></tr></table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Name	Type	Description	
		0 or omitted	US (NASD) 30/360. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, if the first date has a day value of 31, the date is converted to day 30. If the second date has a day value of 31, it is changed to 30 days as long as the first date was not 28 or 29 February, in which case it does not change.
		1	Actual/actual. The actual number of days between the two dates are counted. If the date range includes the date 29 February, the year is 366 days; otherwise it is 365 days.
		2	Actual/360. Similar to Basis 1, but only has 360 days per year.
		3	Actual/365. Similar to Basis 1, but always has 365 days per year.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Name	Type	Description		
			4	European 30/360. The European method for adjusting day counts. Assumes that each month has 30 days and the total number of days in the year is 360 by making the following adjustments: • If the date is 28 or 29 February, it is adjusted to 30 February. • For months with 31 days, all dates with a day value of 31 are changed to day 30, including situations where the first date is 28 or 29 February.

Time information in the date arguments is ignored.

Return Type and Value: number – The annual yield of a security that pays interest at maturity.

However, if

- *settlement*, *maturity*, or *issue* is out of range for the current date system, #NUM! is returned.
- *issue* ≥ *settlement*, #NUM! is returned.
- *settlement* ≥ *maturity*, #NUM! is returned.
- *rate* < 0, #NUM! is returned.
- *pr* ≤ 0, #NUM! is returned.
- *basis* < 0 or *basis* > 4, #NUM! is returned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*:

```
YIELDMAT(DATE(2008,3,15),DATE(2008,11,3),DATE(2007,11,8),0.0625,
  100.0123,0) results in 6.0954%
```

end example]

18.17.7.356 ZTEST

Syntax:

ZTEST (*array* , *test-value* [, *sigma*])

Description: Computes the one-tailed probability-value of a z-test. For a given hypothesized population mean, *test-value*, ZTEST returns the probability that the sample mean would be greater than the average of observations in the data set *array*; that is, the observed sample mean.

Mathematical Formula:

When *sigma* is present:

$$ZTEST(array, \mu_0) = 1 - NORMSDIST\left((\bar{x} - \mu_0)/(sigma/\sqrt{n})\right)$$

When *sigma* is omitted:

$$ZTEST(array, \mu_0) = 1 - NORMSDIST\left((\bar{x} - \mu_0)/(s/\sqrt{n})\right)$$

where:

- n = the number of observations in the sample COUNT(*array*)
- s = the sample standard deviation STDEV(*array*)
- x = a sample value
- $\bar{x}$ = the sample mean AVERAGE(*array*)
- μ_0 = the argument *test-value*

Arguments:

Name	Type	Description
array	array	The set of numerical data against which to test *test-value*.
test-value	number	The number to test.
sigma	number	The number is the population (known) standard deviation. If omitted, the sample standard deviation is used.

Return Type and Value: number – The one-tailed probability-value of a z-test.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

However, if:

- *array* is empty, the return value is unspecified.
- *sigma* is ≤ 0, `#NUM!` is returned.

[*Example*:

`ZTEST({3,6,7,8,6,5,4,2,1,9},4)` results in `0.090574197`
`ZTEST({3,6,7,8,6,5,4,2,1,9},6)` results in `0.863043389`

end example]

18.18 Simple Types

This is the complete list of simple types dedicated to SpreadsheetML.

18.18.1 ST_Axis (PivotTable Axis)

This simple type defines the axes for a PivotTable selection.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
axisCol (Column Axis)	Column axis
axisPage (Include Count Filter)	Page axis
axisRow (Row Axis)	Row axis
axisValues (Values Axis)	Values axis

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Axis) is located in §A.2. *end note*]

18.18.2 ST_BorderId (Border Id)

Zero-based index of the border record used by this cell format.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BorderId) is located in §A.2. *end note*]

18.18.3 ST_BorderStyle (Border Line Styles)

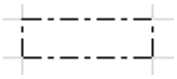
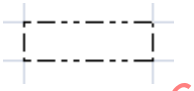
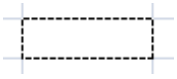
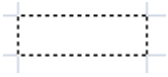
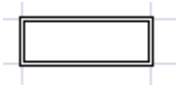
The line style of a border in a cell.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

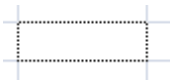
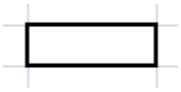
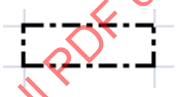
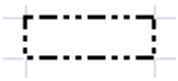
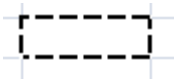
 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dashDot (Dash Dot)	The line style of a border is dash-dot. [*Example*:  *end example*]
dashDotDot (Dash Dot Dot)	The line style of a border is dash-dot-dot. [*Example*:  *end example*]
dashed (Dashed)	The line style of a border is dashed. [*Example*:  *end example*]
dotted (Dotted)	The line style of a border is dotted. [*Example*:  *end example*]
double (Double Line)	The line style of a border is double line. [*Example*:  *end example*]
hair (Hairline Border)	The line style of a border is hairline.

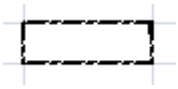
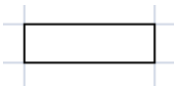
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*:  *end example*]
medium (Medium Border)	The line style of a border is medium. [*Example*:  *end example*]
mediumDashDot (Medium Dash Dot)	The line style of a border is medium dash-dot. [*Example*:  *end example*]
mediumDashDotDot (Medium Dash Dot Dot)	The line style of a border is medium dash-dot-dot. [*Example*:  *end example*]
mediumDashed (Medium Dashed)	The line style of a border is medium dashed. [*Example*:  *end example*]
none (None)	The line style of a border is none (no border visible). [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
slantDashDot (Slant Dash Dot)	The line style of a border is slant-dash-dot. [*Example*:  *end example*]
thick (Thick Line Border)	The line style of a border is 'thick'. [*Example*: *end example*]
thin (Thin Border)	The line style of a border is thin. [*Example*:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BorderStyle) is located in §A.2. *end note*]

18.18.4 ST_CalcMode (Calculation Mode)

This simple type defines the supported modes for performing calculations on workbook data.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatic)	Indicates that calculations in the workbook are performed automatically when cell values change. The application recalculates those cells that are dependent

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	on other cells that contain changed values. This mode of calculation helps to avoid unnecessary calculations.
autoNoTable (Automatic Calculation (No Tables))	Indicates tables be excluded during automatic calculation.
manual (Manual Calculation Mode)	Indicates that calculations in the workbook be triggered manually by the user. For example, the application might expose a command in the user interface.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CalcMode) is located in §A.2. *end note*]

18.18.5 ST_CellComments (Cell Comments)

These enumerations specify how cell comments shall be displayed for paper printing purposes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
asDisplayed (Print Comments As Displayed)	Print cell comments as displayed.
atEnd (Print At End)	Print cell comments at end of document.
none (None)	Do not print cell comments.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellComments) is located in §A.2. *end note*]

18.18.6 ST_CellFormulaType (Formula Type)

Indicates the type of formula in the cell.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
array (Array Formula)	Formula is an array formula.
dataTable (Table Formula)	Formula is a data table formula.
normal (Normal)	Formula is a regular cell formula.
shared (Shared Formula)	Formula is part of a shared formula.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellFormulaType) is located in §A.2. *end note*]

18.18.7 ST_CellRef (Cell Reference)

Represents a single cell reference in a SpreadsheetML document.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellRef) is located in §A.2. *end note*]

18.18.8 ST_CellSpan (Cell Span Type)

A single cell span item.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellSpan) is located in §A.2. *end note*]

18.18.9 ST_CellSpans (Cell Spans)

List of the cell spans of the item.

This simple type allows a list of items of the ST_CellSpan simple type (§18.18.8).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellSpans) is located in §A.2. *end note*]

18.18.10 ST_CellStyleXfId (Cell Style Format Id)

Used by xf records and cellStyle records to reference xf records defined in the cellStyleXfs collection.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellStyleXfId) is located in §A.2. *end note*]

18.18.11 ST_CellType (Cell Type)

Indicates the cell's data type.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Boolean)	Cell containing a boolean.
d (Date)	Cell contains a date in the ISO 8601 format.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
e (Error)	Cell containing an error.
inlineStr (Inline String)	Cell containing an (inline) rich string, i.e., one not in the shared string table. If this cell type is used, then the cell value is in the is element rather than the v element in the cell (c element).
n (Number)	Cell containing a number.
s (Shared String)	Cell containing a shared string.
str (String)	Cell containing a formula string.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CellType) is located in §A.2. *end note*]

18.18.12 ST_CfType (Conditional Format Type)

Conditional format rule type.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aboveAverage (Above or Below Average)	This conditional formatting rule highlights cells that are above or below the average for all values in the range.
beginsWith (Begins With)	This conditional formatting rule highlights cells in the range that begin with the given text. Equivalent to using the LEFT() sheet function and comparing values.
cellIs (Cell Is)	This conditional formatting rule compares a cell value to a formula calculated result, using an operator.
colorScale (Color Scale)	This conditional formatting rule creates a gradated color scale on the cells.
containsBlanks (Contains Blanks)	This conditional formatting rule highlights cells that are completely blank. Equivalent of using LEN(TRIM()). This means that if the cell contains only characters that TRIM() would remove, then it is considered blank. An empty cell is also considered blank.
containsErrors (Contains Errors)	This conditional formatting rule highlights cells with formula errors. Equivalent to using ISERROR() sheet function to determine if there is a formula error.
containsText (Contains Text)	This conditional formatting rule highlights cells containing given text. Equivalent to using the SEARCH() sheet function to determine whether the cell contains

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the text.
dataBar (Data Bar)	This conditional formatting rule displays a gradated data bar in the range of cells.
duplicateValues (Duplicate Values)	This conditional formatting rule highlights duplicated values.
endsWith (Ends With)	This conditional formatting rule highlights cells ending with given text. Equivalent to using the RIGHT() sheet function and comparing values.
expression (Expression)	This conditional formatting rule contains a formula to evaluate. When the formula result is true, the cell is highlighted.
iconSet (Icon Set)	This conditional formatting rule applies icons to cells according to their values.
notContainsBlanks (Contains No Blanks)	This conditional formatting rule highlights cells that are not blank. Equivalent of using LEN(TRIM()). This means that if the cell contains only characters that TRIM() would remove, then it is considered blank. An empty cell is also considered blank.
notContainsErrors (Contains No Errors)	This conditional formatting rule highlights cells without formula errors. Equivalent to using ISERROR() sheet function to determine if there is a formula error.
notContainsText (Does Not Contain Text)	This conditional formatting rule highlights cells that do not contain given text. Equivalent to using the SEARCH() sheet function.
timePeriod (Time Period)	This conditional formatting rule highlights cells containing dates in the specified time period. The underlying value of the cell is evaluated, therefore the cell does not need to be formatted as a date to be evaluated. For example, with a cell containing the value 38913 the conditional format shall be applied if the rule requires a value of 7/14/2006.
top10 (Top 10)	This conditional formatting rule highlights cells whose values fall in the top N or bottom N bracket, as specified.
uniqueValues (Unique Values)	This conditional formatting rule highlights unique values in the range.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CfType) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.18.13 ST_CfvoType (Conditional Format Value Object Type)

This simple type expresses the type of the conditional formatting value object (cfvo). In general the cfvo specifies one value used in the gradated scale (max, min, midpoint, etc).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
formula (Formula)	The minimum/ midpoint / maximum value for the gradient is determined by a formula.
max (Maximum)	Indicates that the maximum value in the range shall be used as the maximum value for the gradient.
min (Minimum)	Indicates that the minimum value in the range shall be used as the minimum value for the gradient.
num (Number)	Indicates that the minimum / midpoint / maximum value for the gradient is specified by a constant numeric value.
percent (Percent)	Value indicates a percentage between the minimum and maximum values in the range shall be used as the minimum / midpoint / maximum value for the gradient.
percentile (Percentile)	Value indicates a percentile ranking in the range shall be used as the minimum / midpoint / maximum value for the gradient.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CfvoType) is located in §A.2. *end note*]

18.18.14 ST_Comments (Comment Display Types)

This simple type defines options for displaying comments in the user interface.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
commIndAndComment (Show Comment & Indicator)	Indicates that both the comment indicator and comment text be show in the user interface.
commIndicator (Show Comment Indicator)	Indicates that only the comment indicator be shown in the user interface.
commNone (No Comments)	Indicates that comments not be shown in the user interface.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Comments) is located in §A.2. *end note*]

18.18.15 ST_ConditionalFormattingOperator (Conditional Format Operators)

These conditional format operators are used for "Highlight Cells That Contain…" rules. For example, "highlight cells that begin with "M2" and contain "Mountain Gear"".

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
beginsWith (Begins With)	'Begins with' operator
between (Between)	'Between' operator
containsText (Contains)	'Contains' operator
endsWith (Ends With)	'Ends with' operator
equal (Equal)	'Equal to' operator
greaterThan (Greater Than)	'Greater than' operator
greaterThanOrEqual (Greater Than Or Equal)	'Greater than or equal to' operator
lessThan (Less Than)	'Less than' operator
lessThanOrEqual (Less Than Or Equal)	'Less than or equal to' operator
notBetween (Not Between)	'Not between' operator
notContains (Does Not Contain)	'Does not contain' operator
notEqual (Not Equal)	'Not equal to' operator

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConditionalFormattingOperator) is located in §A.2. *end note*]

18.18.16 ST_CredMethod (Credentials Method)

Credentials method used for server access.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
integrated (Integrated Authentication)	Integrated authentication.
none (No Credentials)	Use no credentials at all.
prompt (Prompt Credentials)	Prompt for credentials.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
stored (Stored Credentials)	Use stored credentials.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CredMethod) is located in §A.2. *end note*]

18.18.17 ST_DataConsolidateFunction (Data Consolidation Functions)

Data consolidation functions specified by the user and used to consolidate ranges of data.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
average (Average)	The average of the values.
count (Count)	The number of data values. The Count consolidation function works the same as the COUNTA worksheet function.
countNums (CountNums)	The number of data values that are numbers. The Count Nums consolidation function works the same as the COUNT worksheet function.
max (Maximum)	The largest value.
min (Minimum)	The smallest value.
product (Product)	The product of the values.
stdDev (StdDev)	An estimate of the standard deviation of a population, where the sample is a subset of the entire population.
stdDevp (StdDevP)	The standard deviation of a population, where the population is all of the data to be summarized.
sum (Sum)	The sum of the values.
var (Variance)	An estimate of the variance of a population, where the sample is a subset of the entire population.
varp (VarP)	The variance of a population, where the population is all of the data to be summarized.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DataConsolidateFunction) is located in §A.2. *end note*]

18.18.18 ST_DataValidationErrorStyle (Data Validation Error Styles)

The style of data validation error alert.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
information (Information Icon)	This data validation error style uses an information icon in the error alert.
stop (Stop Icon)	This data validation error style uses a stop icon in the error alert.
warning (Warning Icon)	This data validation error style uses a warning icon in the error alert.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DataValidationErrorStyle) is located in §A.2. *end note*]

18.18.19 ST_DataValidationImeMode (Data Validation IME Mode)

These values specify that the IME (input method editor) mode is controlled by data validation.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
disabled (Disabled IME Mode)	IME mode is disabled. Forces the IME control to be disabled when this cell is selected.
fullAlpha (Full-Width Alpha-Numeric IME Mode)	Forces the IME control to be on and in full-width alph-numeric input mode when the cell is first selected.
fullHangul (Full Width Hangul)	Forces the IME control to be on and in full-width Hangul input mode when first selecting the cell. Applies when the application's language is Korean and a Korean IME control is selected.
fullKatakana (Full Katakana IME Mode)	Forces the IME control to be on and in full-width Katakana input mode when first selecting the cell. Applies when the application's language is Japanese and a Japanese IME control is selected.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
halfAlpha (Half Alpha IME)	Forces the IME control to be on and in half-width alph-numeric input mode when the cell is first selected.
halfHangul (Half-Width Hangul IME Mode)	Forces the IME control to be on and in half-width Hangul input mode when first selecting the cell. Applies when the application's language is Korean and a Korean IME control is selected.
halfKatakana (Half-Width Katakana)	Forces the IME control to be on and in half-width Katakana input mode when first selecting the cell. Applies when the application's language is Japanese and a Japanese IME control is selected.
hiragana (Hiragana IME Mode)	Forces the IME control to be on and in Hiragana input mode when first selecting the cell. Applies when the application's language is Japanese and a Japanese IME control is selected
noControl (IME Mode Not Controlled)	Data validation does not control the IME control's mode.
off (IME Off)	Forces the IME control to be off when first selecting the cell (goes to direct cell input mode).
on (IME On)	Forces the IME control to be on when first selecting the cell.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DataValidationImeMode) is located in §A.2. *end note*]

18.18.20 ST_DataValidationOperator (Data Validation Operator)

The relational operator used in data validation.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
between (Between)	Data validation which checks if a value is between two other values.
equal (Equal)	Data validation which checks if a value is equal to a specified value.
greaterThan (Greater Than)	Data validation which checks if a value is greater than a specified value.
greaterThanOrEqual (Greater Than Or Equal)	Data validation which checks if a value is greater than or equal to a specified value.
lessThan (Less Than)	Data validation which checks if a value is less than a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	specified value.
lessThanOrEqual (Less Than Or Equal)	Data validation which checks if a value is less than or equal to a specified value.
notBetween (Not Between)	Data validation which checks if a value is not between two other values.
notEqual (Not Equal)	Data validation which checks if a value is not equal to a specified value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DataValidationOperator) is located in §A.2. *end note*]

18.18.21 ST_DataValidationType (Data Validation Type)

Specifies the type of data validation used to validate user input.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
custom (Custom)	Data validation which uses a custom formula to check the cell value.
date (Date)	Data validation which checks for date values satisfying the given condition.
decimal (Decimal)	Data validation which checks for decimal values satisfying the given condition.
list (List)	Data validation which checks for a value matching one of list of values.
none (None)	No data validation.
textLength (Text Length)	Data validation which checks for text values, whose length satisfies the given condition.
time (Time)	Data validation which checks for time values satisfying the given condition.
whole (Whole Number)	Data validation which checks for whole number values satisfying the given condition.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DataValidationType) is located in §A.2. *end note*]

18.18.22 ST_DateTimeGrouping (Date Time Grouping)

Specifies how to group dateTime values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
day (Day)	Group by day
hour (Group by Hour)	Group by hour
minute (Group by Minute)	Group by minute
month (Month)	Group by month
second (Second)	Group by second
year (Group by Year)	Group by year

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DateTimeGrouping) is located in §A.2. *end note*]

18.18.23 ST_DdeValueType (DDE Value Types)

This simple type indicates the type of the DDE value.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Boolean)	Indicates that the value is a boolean.
e (Error)	Indicates that the value is an error.
n (Real Number)	Indicates that the value is a real number.
nil (Nil)	Indicates that the value is nil.
str (String)	Indicates that the value is a string.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DdeValueType) is located in §A.2. *end note*]

18.18.24 ST_DvAspect (Data View Aspect Type)

Specifies the desired data or view aspect of the object when drawing or getting data.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
DVASPECT_CONTENT (Object Display Content)	Provides a representation of an object so it can be displayed as an embedded object inside of a container.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
DVASPECT_ICON (Object Display Icon)	Provides an iconic representation of an object.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DvAspect) is located in §A.2. *end note*]

18.18.25 ST_DxfId (Format Id)

This simple type defines the identifier to CT_Dxfs in the styles part. This a zero-based index. See §18.8.30 in Style for more information on formats.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DxfId) is located in §A.2. *end note*]

18.18.26 ST_DynamicFilterType (Dynamic Filter)

These are the dynamic filter types. A dynamic filter returns a result set which might vary due to a change in the data itself or a change in the date on which the filter is being applied. For example, for a set of data {1,1,2,3}, the aboveAverage filter would return or highlight the last two values in the set. If the data is refreshed or changed to {1,1,1,2}, then only the last value would be highlighted. Similarly, the meaning of "lastQuarter" shall be the same for the dates in January, February, and March, but shall change meaning once the date advances from March to April.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aboveAverage (Above Average)	Shows values that are above average.
belowAverage (Below Average)	Shows values that are below average.
lastMonth (Last Month)	Shows last month's dates.
lastQuarter (Last Quarter)	Shows last calendar quarter's dates.
lastWeek (Last Week)	Shows last week's dates, using Sunday as the first weekday.
lastYear (Last Year)	Shows last year's dates.
M1 (1st Month)	Shows the dates that are in January, regardless of year.
M10 (10th Month)	Shows the dates that are in October, regardless of year.
M11 (11th Month)	Shows the dates that are in November, regardless of year.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
M12 (12th Month)	Shows the dates that are in December, regardless of year.
M2 (2nd Month)	Shows the dates that are in February, regardless of year.
M3 (3rd Month)	Shows the dates that are in March, regardless of year.
M4 (4th Month)	Shows the dates that are in April, regardless of year.
M5 (5th Month)	Shows the dates that are in May, regardless of year.
M6 (6th Month)	Shows the dates that are in June, regardless of year.
M7 (7th Month)	Shows the dates that are in July, regardless of year.
M8 (8th Month)	Shows the dates that are in August, regardless of year.
M9 (9th Month)	Shows the dates that are in September, regardless of year.
nextMonth (Next Month)	Shows next month's dates.
nextQuarter (Next Quarter)	Shows next calendar quarter's dates.
nextWeek (Next Week)	Shows next week's dates, using Sunday as the first weekday.
nextYear (Next Year)	Shows next year's dates.
null (Null)	Common filter type not available.
Q1 (1st Quarter)	Shows the dates that are in the 1st calendar quarter, regardless of year.
Q2 (2nd Quarter)	Shows the dates that are in the 2nd calendar quarter, regardless of year.
Q3 (3rd Quarter)	Shows the dates that are in the 3rd calendar quarter, regardless of year.
Q4 (4th Quarter)	Shows the dates that are in the 4th calendar quarter, regardless of year.
thisMonth (This Month)	Shows this month's dates.
thisQuarter (This Quarter)	Shows this calendar quarter's dates.
thisWeek (This Week)	Shows this week's dates, using Sunday as the first weekday.
thisYear (This Year)	Shows this year's dates.
today (Today)	Shows today's dates.
tomorrow (Tomorrow)	Shows tomorrow's dates.
yearToDate (Year To Date)	Shows the dates between the beginning of the year and today, inclusive.
yesterday (Yesterday)	Shows yesterday's dates.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DynamicFilterType) is located in §A.2. *end note*]

18.18.27 ST_ExternalConnectionType (Text Field Datatype)

These are the possible data types to use when importing text into the SpreadsheetML document. Strings are converted to these data types in the worksheet.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
DMY (Day Month Year)	Field contains a date in the order: day, month, year.
DYM (Day Year Month)	Field contains a date in the order: day, year, month.
EMD (East Asian Year Month Day)	Field contains an East Asian date in the order: EA era year, month, day.
general (General)	The SpreadsheetML application decides the best fit data type based on the content.
MDY (Month Day Year)	Field contains a date in the order: month, day, year.
MYD (Month Day Year)	Field contains a date in the order: month, year, day.
skip (Skip Field)	Don't import this field at all.
text (Text)	Field contains text.
YDM (Year Day Month)	Field contains a date in the order: year, day, month.
YMD (Year Month Day)	Field contains a date in the order: year, month, day.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ExternalConnectionType) is located in §A.2. *end note*]

18.18.28 ST_FieldSortType (Field Sort Type)

This simple type defines the sort orders that can be applied to fields in a PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ascending (Ascending)	Indicates the field is sorted in ascending order.
descending (Descending)	Indicates the field is sorted in descending order.
manual (Manual Sort)	Indicates the field is sorted manually.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FieldSortType) is located in §A.2. *end note*]

18.18.29 ST_FileType (File Type)

The file type being used for text import.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dos (DOS)	DOS (PC-8).
lin (Linux)	Linux
mac (Macintosh)	Macintosh.
other (Other Non-Specified Values)	Other non-specified values at the time of writing.
win (Windows (ANSI))	Windows (ANSI).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FileType) is located in §A.2. *end note*]

18.18.30 ST_FillId (Fill Id)

Zero-based index used to reference a fill record.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FillId) is located in §A.2. *end note*]

18.18.31 ST_FilterOperator (Filter Operator)

Operator enumerations for filtering.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
equal (Equal)	Show results which are equal to criteria.
greaterThan (Greater Than)	Show results which are greater than criteria.
greaterThanOrEqual (Greater Than Or Equal)	Show results which are greater than or equal to criteria.
lessThan (Less Than)	Show results which are less than criteria.
lessThanOrEqual (Less Than Or Equal)	Show results which are less than or equal to criteria.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
notEqual (Not Equal)	Show results which are not equal to criteria.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FilterOperator) is located in §A.2. *end note*]

18.18.32 ST_FontId (Font Id)

An integer that represents a zero based index into the <fonts> collection in the style sheet.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontId) is located in §A.2. *end note*]

18.18.33 ST_FontScheme (Font scheme Styles)

Defines the font scheme, if any, to which this font belongs.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
major (Major Font)	This font is the major font for this theme.
minor (Minor Font)	This font is the minor font for this theme.
none (None)	This font is not a theme font.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontScheme) is located in §A.2. *end note*]

18.18.34 ST_FormatAction (PivotTable Format Types)

This simple type defines the type of formats that can be applied to PivotTables.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
blank (Blank)	Indicates no format is applied to the PivotTable. This value is used when formatting is cleared from already formatted cells in the PivotTable.
drill (Drill Type)	Indicates the PivotTable has drill-through format.
formatting (Formatting)	Indicates the PivotTable has formatting.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
formula (Formula Type)	Indicates the PivotTable has formulas.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FormatAction) is located in §A.2. *end note*]

18.18.35 ST_Formula (Formula)

A formula

This simple type's contents are a restriction of the ST_Xstring datatype (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Formula) is located in §A.2. *end note*]

18.18.36 ST_FormulaExpression (Formula Expression Type)

This simple type specifies an expression type that can comprise a formula.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
area (Area)	Reference to a range of cells.
areaError (Area Error)	Reference to a range of cells that now evaluates to an error.
computedArea (Computed Area)	Computed area reference.
ref (Reference)	Single cell reference.
refError (Reference Is Error)	Single cell reference that now evaluates to an error.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FormulaExpression) is located in §A.2. *end note*]

18.18.37 ST_GradientType (Gradient Type)

Type of gradient fill being used, either linear or path.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
linear (Linear Gradient)	This gradient fill is of linear gradient type. Linear gradient type means that the transition from one color

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	to the next is along a line (e.g., horizontal, vertical, diagonal, etc.).
path (Path)	This gradient fill is of path gradient type. Path gradient type means the that the boundary of transition from one color to the next is a rectangle, defined by top, bottom, left, and right attributes on the gradientFill element.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GradientType) is located in §A.2. *end note*]

18.18.38 ST_GroupBy (Values Group By)

This simple type defines types of data grouping that can be performed on a PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
days (Days)	Indicates a grouping on "days" for date values.
hours (Hours)	Indicates a grouping on "hours" for date values.
minutes (Minutes)	Indicates a grouping on "minutes" for date values.
months (Months)	Indicates a grouping on "months" for date values.
quarters (Quarters)	Indicates a grouping on "quarters" for date values.
range (Group By Numeric Ranges)	Indicates a grouping by numeric ranges for numeric values.
seconds (Seconds)	Indicates a grouping on "seconds" for date values.
years (Years)	Indicates a grouping on "years" for date values.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GroupBy) is located in §A.2. *end note*]

18.18.39 ST_GrowShrinkType (Grow Shrink Type)

This type enumerates behavior patterns for refreshing external data in a query table.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
insertClear (Insert & Clear On Refresh)	Insert entire rows for new data, clear unused cells.
insertDelete (Insert & Delete On Refresh)	Insert cells for new data, delete unused cells.
overwriteClear (Overwrite & Clear On Refresh)	Overwrite existing cells with new data, clear unused cells.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GrowShrinkType) is located in §A.2. *end note*]

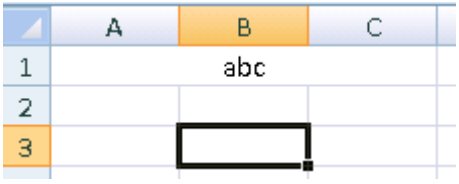
18.18.40 ST_HorizontalAlignment (Horizontal Alignment Type)

The enumeration value indicating the portion of Cell Alignment in a cell format (XF) that is horizontal alignment, i.e., whether it is aligned general, left, right, horizontally centered, filled (replicated), justified, centered across multiple cells, or distributed.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

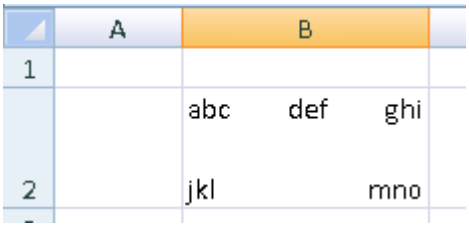
This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Centered Horizontal Alignment)	The horizontal alignment is centered, meaning the text is centered across the cell.

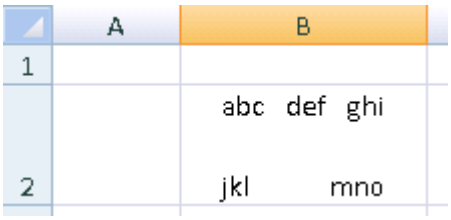
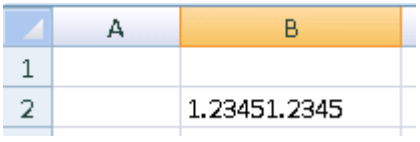
centerContinuous (Center Continuous Horizontal Alignment)	The horizontal alignment is centered across multiple cells. The information about how many cells to span is expressed in the Sheet Part, in the row of the cell in question. For each cell that is spanned in the alignment, a cell element needs to be written out, with the same style Id which references the `centerContinuous` alignment.
	
	[*Example*:
	This shows the value of A1 centered across A1:C1:
	 
	
	The XML from the Sheet Part:
	
	`<row r="1" spans="1:3">`
	`<c r="A1" s="1" t="s">`
	`<v>0</v>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

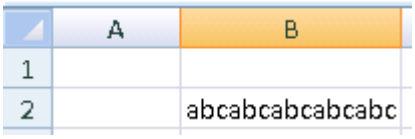
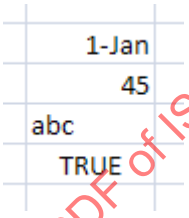
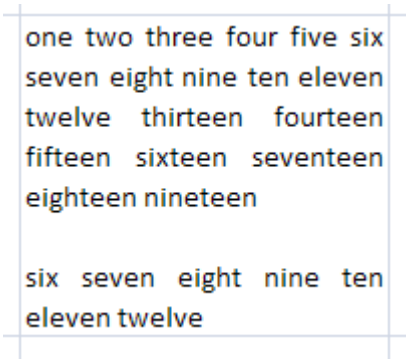
Enumeration Value	Description
	</c> <c r="B1" s="1"/> <c r="C1" s="1"/> </row> The XML from the Styles Part: <cellXfs count="2"> <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"/> <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0" applyAlignment="1"> <alignment horizontal="centerContinuous"/> </xf> </cellXfs> *end example*]
distributed (Distributed Horizontal Alignment)	I/ndicates that each 'word' in each line of text inside the cell is evenly distributed across the width of the cell, with flush right and left margins. When there is also an indent value to apply, both the left and right side of the cell are padded by the indent value. A 'word' is a set of characters with no space character in them. Two lines inside a cell are separated by a carriage return. [*Example*: This shows three lines of text evenly distributed horizontally across the cell. The first line is "abc def ghi", the second line is blank, and the third line is "jkl mno".  This shows the same example, with an indent value of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	2:  There is no vertical component to the alignment being shown here. The row has been manually adjusted to display the text. *end example*]
fill (Fill)	Indicates that the value of the cell should be filled across the entire width of the cell. If blank cells to the right also have the fill alignment, they are also filled with the value, using a convention similar to `centerContinuous`. Additional rules: • Only whole values can be appended, not partial values. • The column will not be widened to 'best fit' the filled value • If appending an additional occurrence of the value exceeds the boundary of the cell left/right edge, don't append the additional occurrence of the value. • The display value of the cell is filled, not the underlying raw number. [*Example*: This cell is filled with the value 1.2345 and has a width of 15 characters:  This cell is filled with the value abc and has width of 15 characters:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 • *end example*] •
general (General Horizontal Alignment)	The horizontal alignment is general-aligned. Text data is left-aligned. Numbers, dates, and times are right-aligned. Boolean types are centered. Changing the alignment does not change the type of data. [*Example*: These cells are `general` aligned:  *end example*]
justify (Justify)	The horizontal alignment is justified (flush left and right). For each line of text, aligns each line of the wrapped text in a cell to the right and left (except the last line). If no single line of text wraps in the cell, then the text is not justified. [*Example*: There are two lines of text in this cell, and the cell's horizontal alignment is `justify`:  *end example*]
left (Left Horizontal Alignment)	The horizontal alignment is left-aligned, even in Right-

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	to-Left mode. Aligns contents at the left edge of the cell. If an indent amount is specified, the contents of the cell is indented from the left by the specified number of character spaces. The character spaces are based on the default font and font size for the workbook.
right (Right Horizontal Alignment)	The horizontal alignment is right-aligned, meaning that cell contents are aligned at the right edge of the cell, even in Right-to-Left mode.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HorizontalAlignment) is located in §A.2. *end note*]

18.18.41 ST_HtmlFmt (HTML Formatting Handling)

How to handle formatting from the HTML source.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Transfer all HTML formatting into the worksheet along with data.
none (No Formatting)	Bring data in as unformatted text (setting data types still occurs).
rtf (Honor Rich Text)	Translate HTML formatting to rich text formatting on the data brought into the worksheet.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HtmlFmt) is located in §A.2. *end note*]

18.18.42 ST_IconSetType (Icon Set Type)

Icon set type for conditional formatting. The threshold values for triggering the different icons within a set are configurable, and the icon order is reversible. See element iconSet for more information.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
3Arrows (3 Arrows)	3 arrows icon set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
3ArrowsGray (3 Arrows (Gray))	3 gray arrows icon set.
3Flags (3 Flags)	3 flags icon set.
3Signs (3 Signs)	3 signs icon set.
3Symbols (3 Symbols Circled)	3 symbols icon set.
3Symbols2 (3 Symbols)	3 Symbols icon set.
3TrafficLights1 (3 Traffic Lights)	3 traffic lights icon set (#1).
3TrafficLights2 (3 Traffic Lights Black)	3 traffic lights icon set with thick black border.
4Arrows (4 Arrows)	4 arrows icon set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
4ArrowsGray (4 Arrows (Gray))	4 gray arrows icon set.
4Rating (4 Ratings)	4 ratings icon set.
4RedToBlack (4 Red To Black)	4 'red to black' icon set.
4TrafficLights (4 Traffic Lights)	4 traffic lights icon set.
5Arrows (5 Arrows)	5 arrows icon set.
5ArrowsGray (5 Arrows (Gray))	5 gray arrows icon set.
5Quarters (5 Quarters)	5 quarters icon set.
5Rating (5 Ratings Icon Set)	5 rating icon set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_IconSetType) is located in §A.2. *end note*]

18.18.43 ST_ItemType (PivotItem Type)

This simple type defines the pivot type for a pivotItem.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
avg (Average)	Indicates the pivot item represents an "average" aggregate function.
blank (Blank Pivot Item)	Indicates the pivot item represents a blank line.
count (Count)	Indicates the pivot item represents custom the "count" aggregate."
countA (CountA)	Indicates the pivot item represents the "count numbers" aggregate function.
data (Data)	Indicate the pivot item represents data.
default (Default)	Indicates the pivot item represents the default type for this PivotTable. The default pivot item type is the "total" aggregate function.
grand (Grand Total Item)	Indicates the pivot items represents the grand total line.
max (Max)	Indicates the pivot item represents the "maximum" aggregate function.
min (Min)	Indicates the pivot item represents the "minimum" aggregate function.
product (Product)	Indicates the pivot item represents the "product" function.
stdDev (stdDev)	Indicates the pivot item represents the "standard deviation" aggregate function.
stdDevP (StdDevP)	Indicates the pivot item represents the "standard deviation population" aggregate function.
sum (Sum)	Indicates the pivot item represents the "sum" aggregate value.
var (Var)	Indicates the pivot item represents the "variance" aggregate value.
varP (VarP)	Indicates the pivot item represents the "variance population" aggregate value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ItemType) is located in §A.2. *end note*]

18.18.44 ST_MdxFunctionType (MDX Function Type)

This simple type is an enumeration representing different MDX function types.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
c (Cube Set Count)	CUBESETCOUNT
k (Cube KPI Member)	CUBEKPIMEMBER
m (Cube Member)	CUBEMEMBER
p (Cube Member Property)	CUBEMEMBERPROPERTY
r (Cube Ranked Member)	CUBERANKEDMEMBER
s (Cube Set)	CUBESET
v (Cube Value)	CUBEVALUE

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MdxFunctionType) is located in §A.2. *end note*]

18.18.45 ST_MdxKPIProperty (MDX KPI Property)

An enumeration representing the different types of KPI properties.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
g (Goal)	Goal.
m (Current Time Member)	Current time member.
s (Status)	Status.
t (Trend)	Trend.
v (Value)	Value.
w (Weight)	Weight.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MdxKPIProperty) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.18.46 ST_MdxSetOrder (MDX Set Order)

This simple type represents an enumeration specifying an MDX set order.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
a (Ascending)	Sort ascending.
aa (Alpha Ascending Sort Order)	Sorted alphabetically in ascending order by the caption.
ad (Alpha Descending Sort Order)	Sort in descending order alphabetically by the caption.
d (Descending)	Sort descending.
na (Natural Ascending)	Sorted in ascending order by the natural order of the data - usually by the key. For instance if there is a list of accounts in a general ledger, this might be in order of account number.
nd (Natural Descending)	Sorted in descending order by the natural order of the data - usually by the key. For instance if there is a list of accounts in a general ledger, this might be in order of account number.
u (Unsorted)	Unsorted.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MdxSetOrder) is located in §A.2. *end note*]

18.18.47 ST_NumFmtId (Number Format Id)

This simple type defines the identifier to a style sheet number format entry in CT_NumFmts. Number formats are written to the styles part. See §18.8.31 for more information on number formats.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NumFmtId) is located in §A.2. *end note*]

18.18.48 ST_Objects (Object Display Types)

This simple type defines how the application displays objects in this workbook. Objects might include charts, images, and other object data that the application supports.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Indicates that all objects be shown in the workbook.
none (None)	Indicates that all objects be hidden in the workbook.
placeholders (Show Placeholders)	Indicates that the application show placeholders for objects in the workbook.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Objects) is located in §A.2. *end note*]

18.18.49 ST_OleUpdate (OLE Update Types)

Indicates whether the linked object updates the cached data for the linked object automatically or only when the container calls IOleObject::Update or IOleLink::Update methods.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
OLEUPDATE_ALWAYS (Always Update OLE)	Update the link object whenever possible, this option corresponds to the 'automatic update' option in the Links dialog box.
OLEUPDATE_ONCALL (Update OLE On Call)	Update the link object only when IOleObject::Update or IOleLink::Update is called, this option coresponds to the Manual update option in the Links dialog box.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OleUpdate) is located in §A.2. *end note*]

18.18.50 ST_Orientation (Orientation)

Print orientation for this sheet.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (Default)	Orientation not specified, use the default.
landscape (Landscape)	Landscape orientation.
portrait (Portrait)	Portrait orientation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Orientation) is located in §A.2. *end note*]

18.18.51 ST_PageOrder (Page Order)

Specifies printed page order.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
downThenOver (Down Then Over)	Order pages vertically first, then move horizontally.
overThenDown (Over Then Down)	Order pages horizontally first, then move vertically

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageOrder) is located in §A.2. *end note*]

18.18.52 ST_Pane (Pane Types)

Defines the names of the four possible panes into which the view of a workbook in the application can be split.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottomLeft (Bottom Left Pane)	Bottom left pane, when both vertical and horizontal splits are applied. This value is also used when only a horizontal split has been applied, dividing the pane into upper and lower regions. In that case, this value specifies the bottom pane.
bottomRight (Bottom Right Pane)	Bottom right pane, when both vertical and horizontal splits are applied.
topLeft (Top Left Pane)	Top left pane, when both vertical and horizontal splits are applied. This value is also used when only a horizontal split has been applied, dividing the pane into upper and lower regions. In that case, this value specifies the top pane. This value is also used when only a vertical split has been applied, dividing the pane into right and left regions. In that case, this value specifies the left pane

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
topRight (Top Right Pane)	Top right pane, when both vertical and horizontal splits are applied. This value is also used when only a vertical split has been applied, dividing the pane into right and left regions. In that case, this value specifies the right pane.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Pane) is located in §A.2. *end note*]

18.18.53 ST_PaneState (Pane State)

State of the sheet's pane.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
frozen (Frozen)	Panes are frozen, but were not split being frozen. In this state, when the panes are unfrozen again, a single pane results, with no split. In this state, the split bars are not adjustable.
frozenSplit (Frozen Split)	Panes are frozen and were split before being frozen. In this state, when the panes are unfrozen again, the split remains, but is adjustable.
split (Split)	Panes are split, but not frozen. In this state, the split bars are adjustable by the user.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PaneState) is located in §A.2. *end note*]

18.18.54 ST_ParameterType (Parameter Type)

Parameter Type.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cell (Parameter From Cell)	Get the parameter value from a cell on each refresh.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
prompt (Prompt on Refresh)	Prompt the user on each refresh for a parameter value.
value (Value)	Use a constant value on each refresh for the parameter value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ParameterType) is located in §A.2. *end note*]

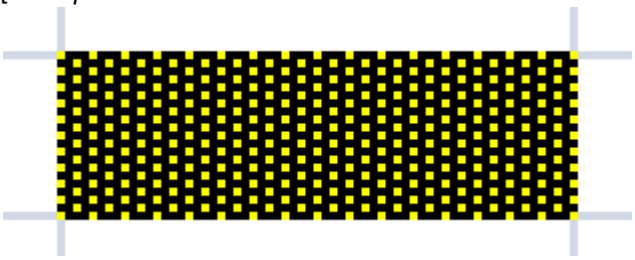
18.18.55 ST_PatternType (Pattern Type)

Indicates the style of fill pattern being used for a cell format.

The examples below use yellow background and black foreground colors.

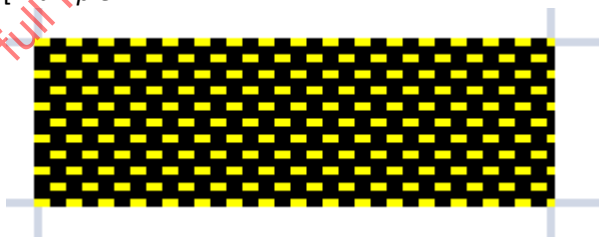
This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
darkDown (Dark Down)	The fill style is 'dark down'. [*Example*: *end example*]
darkGray (Dary Gray)	The fill style is 'dark gray'. [*Example*:  *end example*]
darkGrid (Dark Grid)	The fill style is 'dark grid'. [*Example*:

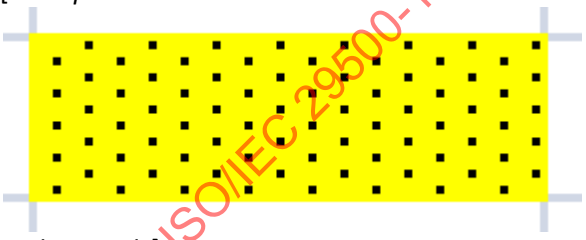
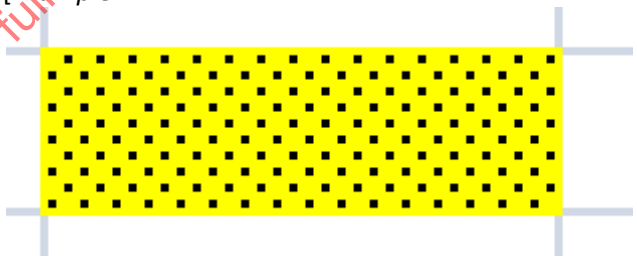
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
darkHorizontal (Dark Horizontal)	The fill style is dark horizontal. [*Example*: *end example*]
darkTrellis (Dark Trellis)	The fill style is 'dark trellis'. [*Example*:  *end example*]
darkUp (Dark Up)	The fill style is 'dark up'. [*Example*: *end example*]
darkVertical (Dark Vertical)	The fill style is 'dark vertical'. [*Example*:

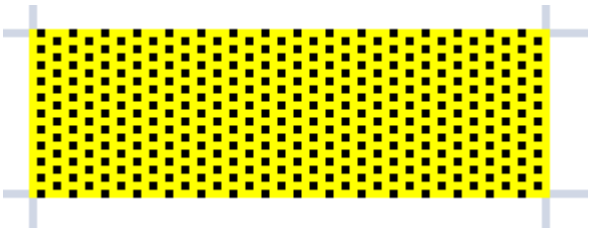
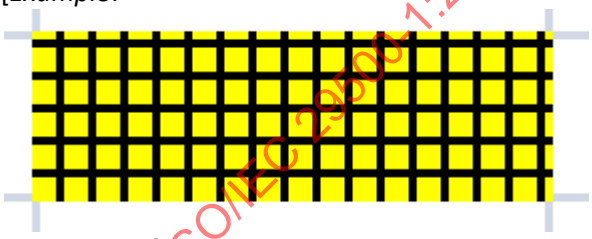
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
gray0625 (Gray 0.0625)	The fill style is grayscale of 0.0625 (1/16) value. [*Example*:  *end example*]
gray125 (Gray 0.125)	The fill style is grayscale of 0.125 (1/8) value. [*Example*:  *end example*]
lightDown (Light Down)	The fill style is 'light down'. [*Example*: *end example*]
lightGray (Light Gray)	The fill style is light gray. [*Example*:

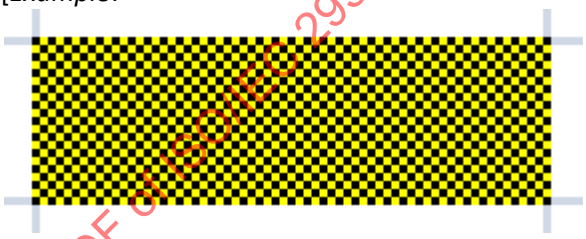
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
lightGrid (Light Grid)	The fill style is 'light grid'. [*Example*:  *end example*]
lightHorizontal (Light Horizontal)	The fill style is light horizontal. [*Example*: *end example*]
lightTrellis (Light Trellis)	The fill style is 'light trellis'. [*Example*: *end example*]
lightUp (Light Up)	The fill style is light up. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
lightVertical (Light Vertical)	The fill style is light vertical.
mediumGray (Medium Gray)	The fill style is medium gray. [*Example*:  *end example*]
none (None)	The fill style is none (no fill). When foreground and/or background colors are specified, a pattern of 'none' overrides and means the cell has no fill. [*Example*: *end example*]
solid (Solid)	The fill style is solid. When solid is specified, the foreground color (fgColor) is the only color rendered, even when a background color (bgColor) is also specified. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	end example]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PatternType) is located in §A.2. *end note*]

18.18.56 ST_PhoneticAlignment (Phonetic Alignment Types)

Phonetic alignment settings. These specify how to align the phonetic text, which represent the sounds, above the base text or base word.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center Alignment)	Center the phonetic characters over the base word, per word.
distributed (Distributed)	Each phonetic character is distributed above each base word character, per word.
left (Left Alignment)	Each phonetic character is left justified with respect to the base text., per word.
noControl (No Control)	Each phonetic character is left justified without respect to the base text (so it is not per word).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PhoneticAlignment) is located in §A.2. *end note*]

18.18.57 ST_PhoneticType (Phonetic Type)

Represents the different East Asian character sets that shall be used for displaying phonetic hints.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fullwidthKatakana (Full-Width Katakana)	Full-width Katakana is used

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
halfwidthKatakana (Half-Width Katakana)	Half-width Katakana is used, this is the same Katakana character set, just half as wide so it takes up less space.
Hiragana (Hiragana)	Hiragana is used
noConversion (No Conversion)	Any characters are allowed. In this case the spreadsheet application shall leave the text as entered.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PhoneticType) is located in §A.2. *end note*]

18.18.58 ST_PivotAreaType (Rule Type)

Indicates the type of rule being used to describe an area or aspect of the PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Refers to the whole PivotTable.
button (Field Button)	Refers to a field button.
data (Data)	Refers to something in the data area.
none (None)	Refers to no Pivot area.
normal (Normal)	Refers to a header or item.
origin (Origin)	Refers to the blank cells at the top-left of the PivotTable (top-left to LTR sheets, top-right for RTL sheets).
topEnd (Top End)	Refers to the blank cells at the top of the PivotTable, on its trailing edge (top-right for LTR sheets, top-left for RTL sheets).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PivotAreaType) is located in §A.2. *end note*]

18.18.59 ST_PivotFilterType (Pivot Filter Types)

This simple type defines filters that can be applied to PivotTables.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
captionBeginsWith (Caption Begins With)	Indicates the "begins with" filter for field captions.
captionBetween (Caption Is Between)	Indicates the "is between" filter for field captions.
captionContains (Caption Contains)	Indicates the "contains" filter for field captions.
captionEndsWith (Caption Ends With)	Indicates the "ends with" filter for field captions.
captionEqual (Caption Equals)	Indicates the "equal" filter for field captions.
captionGreaterThan (Caption Is Greater Than)	Indicates the "is greater than" filter for field captions.
captionGreaterThanOrEqual (Caption Is Greater Than Or Equal To)	Indicates the "is greater than or equal to" filter for field captions.
captionLessThan (Caption Is Less Than)	Indicates the "is less than" filter for field captions.
captionLessThanOrEqual (Caption Is Less Than Or Equal To)	Indicates the "is less than or equal to" filter for field captions.
captionNotBeginsWith (Caption Does Not Begin With)	Indicates the "does not begin with" filter for field captions.
captionNotBetween (Caption Is Not Between)	Indicates the "is not between" filter for field captions.
captionNotContains (Caption Does Not Contain)	Indicates the "does not contain" filter for field captions.
captionNotEndsWith (Caption Does Not End With)	Indicates the "does not end with" filter for field captions.
captionNotEqual (Caption Not Equal)	Indicates the "not equal" filter for field captions.
count (Count)	Indicates the "count" filter.
dateBetween (Date Between)	Indicates the "between" filter for date values.
dateEqual (Date Equals)	Indicates the "equals" filter for date values.
dateNewerThan (Date Newer Than)	Indicates the "newer than" filter for date values.
dateNewerThanOrEqual (Date Newer Than or Equal To)	Indicates the "newer than or equal to" filter for date values.
dateNotBetween (Date Not Between)	Indicates the "not between" filter for date values.
dateNotEqual (Date Does Not Equal)	Indicates the "does not equal" filter for date values.
dateOlderThan (Date Older Than)	Indicates the "older than" filter for date values.
dateOlderThanOrEqual (Date Older Than Or Equal)	Indicates the "older than or equal to" filter for date values.
lastMonth (Last Month)	Indicates the "last month" filter for date values.
lastQuarter (Last Quarter)	Indicates the "last quarter" filter for date values.
lastWeek (Last Week)	Indicates the "last week" filter for date values.
lastYear (Last Year)	Indicates the "last year" filter for date values.
M1 (January)	Indicates the "January" filter for date values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
M10 (Dates in October)	Indicates the "October" filter for date values.
M11 (Dates in November)	Indicates the "November" filter for date values.
M12 (Dates in December)	Indicates the "December" filter for date values.
M2 (Dates in February)	Indicates the "February" filter for date values.
M3 (Dates in March)	Indicates the "March" filter for date values.
M4 (Dates in April)	Indicates the "April" filter for date values.
M5 (Dates in May)	Indicates the "May" filter for date values.
M6 (Dates in June)	Indicates the "June" filter for date values.
M7 (Dates in July)	Indicates the "July" filter for date values.
M8 (Dates in August)	Indicates the "August" filter for date values.
M9 (Dates in September)	Indicates the "September" filter for date values.
nextMonth (Next Month)	Indicates the "next month" filter for date values.
nextQuarter (Next Quarter)	Indicates the "next quarter" for date values.
nextWeek (Next Week)	Indicates the "next week" for date values.
nextYear (Next Year)	Indicates the "next year" filter for date values.
percent (Percent)	Indicates the "percent" filter for numeric values.
Q1 (First Quarter)	Indicates the "first quarter" filter for date values.
Q2 (Second Quarter)	Indicates the "second quarter" filter for date values.
Q3 (Third Quarter)	Indicates the "third quarter" filter for date values.
Q4 (Fourth Quarter)	Indicates the "fourth quarter" filter for date values.
sum (Sum)	Indicates the "sum" filter for numeric values.
thisMonth (This Month)	Indicates the "this month" filter for date values.
thisQuarter (This Quarter)	Indicates the "this quarter" filter for date values.
thisWeek (This Week)	Indicates the "this week" filter for date values.
thisYear (This Year)	Indicate the "this year" filter for date values.
today (Today)	Indicates the "today" filter for date values.
tomorrow (Tomorrow)	Indicates the "tomorrow" filter for date values.
unknown (Unknown)	Indicates the PivotTable filter is unknown to the application.
valueBetween (Value Between)	Indicates the "Value between" filter for text and numeric values.
valueEqual (Value Equal)	Indicates the "value equal" filter for text and numeric values.
valueGreaterThan (Value Greater Than)	Indicates the "value greater than" filter for text and numeric values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
valueGreaterThanOrEqual (Value Greater Than Or Equal To)	Indicates the "value greater than or equal to" filter for text and numeric values.
valueLessThan (Value Less Than)	Indicates the "value less than" filter for text and numeric values.
valueLessThanOrEqual (Value Less Than Or Equal To)	Indicates the "value less than or equal to" filter for text and numeric values
valueNotBetween (Value Not Between)	Indicates the "value not between" filter for text and numeric values.
valueNotEqual (Value Not Equal)	Indicates the "value not equal" filter for text and numeric values.
yearToDate (Year-To-Date)	Indicates the "year-to-date" filter for date values.
yesterday (Yesterday)	Indicates the "yesterday" filter for date values.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PivotFilterType) is located in §A.2. *end note*]

18.18.60 ST_PrintError (Print Errors)

This enumeration specifies how to display cells with errors when printing the worksheet.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
blank (Show Cell Errors As Blank)	Display cell errors as blank.
dash (Dash Cell Errors)	Display cell errors as dashes.
displayed (Display Cell Errors)	Display cell errors as displayed on screen.
NA (NA)	Display cell errors as #N/A.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PrintError) is located in §A.2. *end note*]

18.18.61 ST_Qualifier (Qualifier)

Qualifier to use to denote string data types in when text is imported from an external file.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
doubleQuote (Double Quote)	Quotation mark -- double quote (").
none (No Text Qualifier)	No text string qualifier used.
singleQuote (Single Quote)	Apostrophe mark -- single quote (').

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Qualifier) is located in §A.2. *end note*]

18.18.62 ST_Ref (Cell References)

This simple type defines a reference to a range of cells within a sheet in the workbook. A reference identifies a cell or a range of cells on a worksheet and tells the application where to look for the values or data you want to use in a formula. With references, you can use data contained in different parts of a worksheet in one formula or use the value from one cell in several formulas. You can also refer to cells on other sheets in the same workbook, and to other workbooks. References to cells in other workbooks are called links.

SpreadsheetML defines two reference styles defined in the ST_RefMode simple type.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Ref) is located in §A.2. *end note*]

18.18.63 ST_RefA (Single Cell Reference)

This simple type specifies a single cell reference that might be absolute.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RefA) is located in §A.2. *end note*]

18.18.64 ST_RefMode (Reference Mode)

This simple type defines the supported reference styles or modes for a workbook in SpreadsheetML.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
A1 (A1 Mode)	Indicates that the workbook uses A1 reference style. This is the default for SpreadsheetML. A1 reference style refers to columns with letters and refers to rows with numbers. For example, A1 refers to the cell at the intersection of column A and row 1.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
R1C1 (R1C1 Reference Mode)	Indicates that the workbook uses the R1C1 reference style. R1C1 reference style refers to both the rows and the columns on the worksheet with numbers. The location of a cell is indicated with an "R" followed by a row number and a "C" followed by a column number. For example, R1C1 refers to the cell at the intersection of row R1 and column C1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RefMode) is located in §A.2. *end note*]

18.18.65 ST_RevisionAction (Revision Action Types)

Identifies what kind of action the user performed. Applies to Comment and Custom View revision record.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
add (Add)	Add action.
delete (Delete)	Delete action.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RevisionAction) is located in §A.2. *end note*]

18.18.66 ST_rwColActionType (Row Column Action Type)

Identifies what kind of an action was applied to a row or column.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
deleteCol (Delete Column)	Column delete revision.
deleteRow (Delete Row)	Row delete revision.
insertCol (Column Insert)	Column insert revision.
insertRow (Insert Row)	Row insert revision.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_rwColActionType) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

18.18.67 ST_Scope (Conditional Formatting Scope)

This simple type defines the scope of conditional formatting applied in the PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
data (Data Fields)	Indicates that conditional formatting is applied to the selected data fields.
field (Field Intersections)	Indicates that conditional formatting is applied to the selected PivotTable field intersections.
selection (Selection)	Indicates that conditional formatting is applied to the selected cells.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Scope) is located in §A.2. *end note*]

18.18.68 ST_SheetState (Sheet Visibility Types)

This simple type defines the possible states for sheet visibility.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hidden (Hidden)	Indicates the workbook window is hidden, but can be shown by the user via the user interface.
veryHidden (Very Hidden)	Indicates the sheet is hidden and cannot be shown in the user interface (UI). This state is only available programmatically.
visible (Visible)	Indicates the sheet is visible.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SheetState) is located in §A.2. *end note*]

18.18.69 ST_SheetViewType (Sheet View Type)

This simple type defines the kinds of view available to an application when rendering a SpreadsheetML document. Those view kinds are, as follows: *normal view*, *page break preview*, and *page layout view*.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
normal (Normal View)	Specifies that the worksheet should be displayed without regard to pagination. [*Note*: The worksheet might be displayed as a continuous grid of cells, for example. *end note*]
pageBreakPreview (Page Break Preview)	Specifies that the worksheet should be displayed showing where pages would break if it were printed. [*Note*: The worksheet might be displayed as a continuous grid of cells with lines overlaid, indicating page breaks, for example. *end note*]
pageLayout (Page Layout View)	Specifies that the worksheet should be displayed mimicking how it would look if printed. [*Note*: The worksheet might be displayed as a set of printed pages, showing a discontinuous grid of cells with page borders and margins, for example. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SheetViewType) is located in §A.2. *end note*]

18.18.70 ST_ShowDataAs (Show Data As)

This simple type defines the data formats for a field in the PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
difference (Difference)	Indicates the field is shown as the "difference from" a value.
index (Index)	Indicates the field is shown as the "index.
normal (Normal Data Type)	Indicates that the field is shown as its normal data type.
percent (Percentage Of)	Indicates the field is show as the "percentage of
percentDiff (Percentage Difference)	Indicates the field is shown as the "percentage difference from" a value.
percentOfCol (Percent of Column)	Indicates the field is shown as the percentage of column.
percentOfRow (Percentage of Row)	Indicates the field is shown as the percentage of row
percentOfTotal (Percentage of Total)	Indicates the field is shown as percentage of total.
runTotal (Running Total)	Indicates the field is shown as running total in the table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShowDataAs) is located in §A.2. *end note*]

18.18.71 ST_SmartTagShow (Smart Tag Display Types)

This simple type defines options for displaying smart tags in the user interface.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Indicates that smart tags are enabled and shown in the user interface.
noIndicator (No Smart Tag Indicator)	Indicates that the smart tags are enabled but the indicator not be shown in the user interface.
none (None)	Indicates that smart tags are disabled and not displayed in the user interface.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SmartTagShow) is located in §A.2. *end note*]

18.18.72 ST_SortBy (Sort By)

Specifies what to sort by. In many cases a range of cells are sorted by their values. However, cells can also be sorted by their background color, font color, and type of icon in the cell.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cellColor (Sort by Cell Color)	Sort by cell color
fontColor (Sort by Font Color)	Sort by font color
icon (Sort by Icon)	Sort by icon
value (Value)	Sort by value

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SortBy) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

18.18.73 ST_SortMethod (Sort Method)

Sort method. Chinese Simplified, Chinese Traditional, and Japanese support alternate sort methods (multiple sort options are available). All other languages support only 1 sort option. In that case, the value `pinYin` is used.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (None)	Not specified, use default sort method.
pinYin (PinYin Sort)	Default sort method. This is the only sort option for most languages. For Chinese Simplified, Chinese Traditional, and Japanese, `pinYin` means sort by phonetic value. •
stroke (Sort by Stroke)	Sort by stroke count of the characters. Only applies to Chinese Simplified, Chinese Traditional, and Japanese.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SortMethod) is located in §A.2. *end note*]

18.18.74 ST_SortType (Set Sort Order)

This simple type defines the possible sort order for the PivotTable.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ascending (Ascending)	Indicates that the PivotTable data is sorted in ascending order.
ascendingAlpha (Ascending Alpha)	Indicates that the PivotTable data is sorted in alphabetic order with ascending values.
ascendingNatural (Ascending Natural)	Indicates that the PivotTable data is sorted in natural order with ascending.
descending (Descending)	Indicates that the PivotTable data is sorted in descending.
descendingAlpha (Alphabetic Order Descending)	Indicates that the PivotTable data is sorted in alphabetic order with descending values.
descendingNatural (Natural Order Descending)	Indicates that the PivotTable data is sorted in natural order with descending values.
none (None)	Indicates that the PivotTable data is not sorted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SortType) is located in §A.2. *end note*]

18.18.75 ST_SourceType (PivotCache Type)

This simple type defines the cache types for PivotTables.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
consolidation (Consolidation Ranges)	Indicates that the cache contains data that consolidates ranges.
external (External)	Indicates that the cache contains data from an external data source.
scenario (Scenario Summary Report)	Indicates that the cache contains a scenario summary report
worksheet (Worksheet)	Indicates that the cache contains worksheet data.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SourceType) is located in §A.2. *end note*]

18.18.76 ST_Sqref (Reference Sequence)

A sequence of cell references, space delimited.

This simple type allows a list of items of the ST_Ref simple type (§18.18.62).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Sqref) is located in §A.2. *end note*]

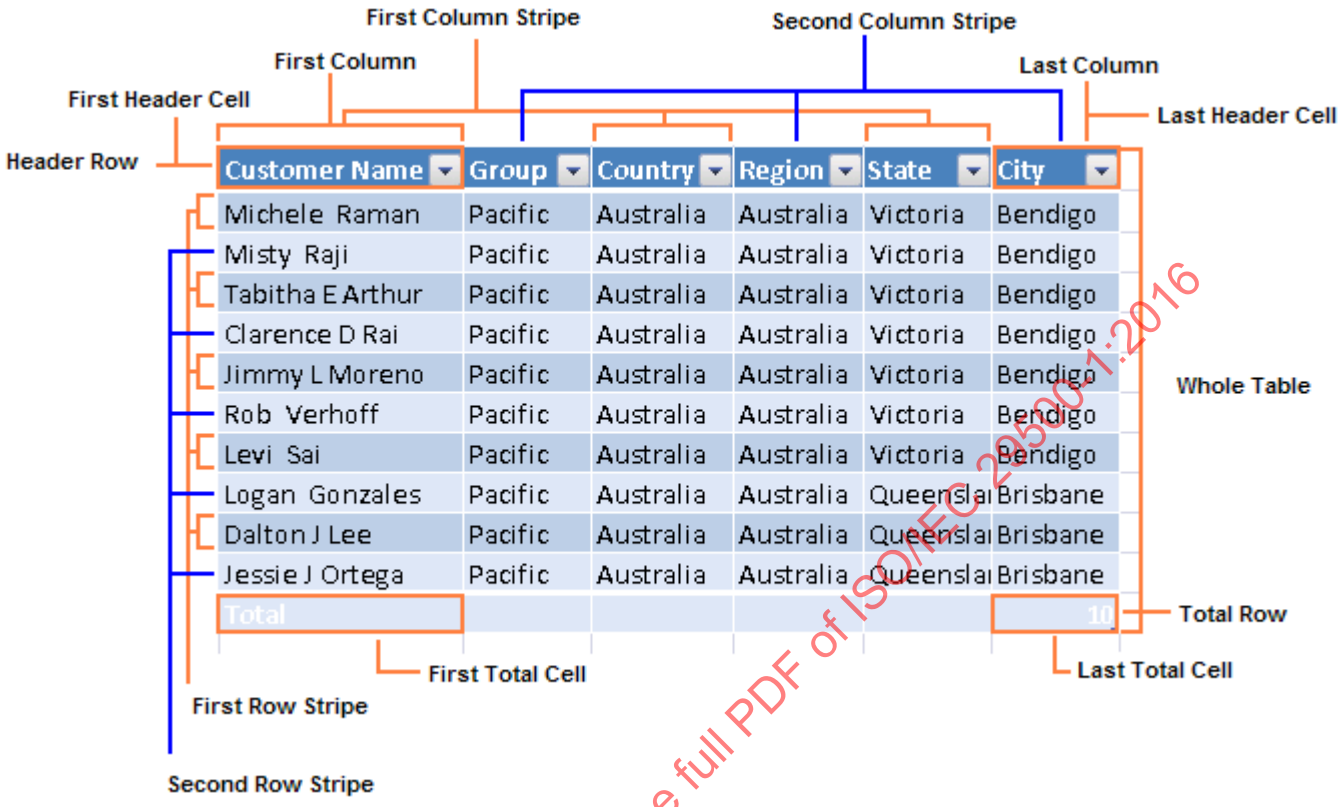
18.18.77 ST_TableStyleType (Table Style Type)

Enumeration of the different structured regions of a Table or PivotTable which can be formatted. Specifies which region is being formatted by this table style element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Table Regions



PivotTable Regions

Blank Row

Only applies when "Insert blank row after each item" is ON.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Country	(All)	
Sum of Sales Amount	Column Labels	
	2001	
	3	
	July	July Total
Row Labels	No Discount	
New South Wales	75625.4664	75625.4664
Coffs Harbour	699.0982	699.0982
Bikes	699.0982	699.0982
Road Bikes	699.0982	699.0982
Road-150 Red, 48		
Road-150 Red, 52		
Road-150 Red, 62		
Road-650 Red, 44	699.0982	699.0982
Darlinghurst	3578.27	3578.27
Bikes	3578.27	3578.27
Road Bikes	3578.27	3578.27
Road-150 Red, 48		
Road-150 Red, 56	3578.27	3578.27
Road-150 Red, 62		
Goulburn	18412.1682	18412.1682
Bikes	18412.1682	18412.1682
Mountain Bikes	3399.99	3399.99
Mountain-100 Silver, 44	3399.99	3399.99
Road Bikes	15012.1782	15012.1782

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Whole Table

Country	(All)	
Sum of Sales Amount	Column Labels	
	2001	
	3	
	July	July Total
Row Labels	No Discount	
New South Wales	75625.4664	75625.4664
Coffs Harbour	699.0982	699.0982
Bikes	699.0982	699.0982
Road Bikes	699.0982	699.0982
Road-150 Red, 48		
Road-150 Red, 52		
Road-150 Red, 62		
Road-650 Red, 44	699.0982	699.0982
Darlinghurst	3578.27	3578.27
Bikes	3578.27	3578.27
Road Bikes	3578.27	3578.27
Road-150 Red, 48		
Road-150 Red, 56	3578.27	3578.27
Road-150 Red, 62		

Page Field Labels

Country	(All)	
Sum of Sales Amount	Column Labels	
	2001	
	3	
	July	July Total
Row Labels	No Discount	
New South Wales	75625.4664	75625.4664
Coffs Harbour	699.0982	699.0982
Bikes	699.0982	699.0982
Road Bikes	699.0982	699.0982

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Page Field Values

Country	(All)		
Sum of Sales Amount	Column Labels		
	2001		
	3		
	July	July Total	
Row Labels	No Discount		
New South Wales	75625.4664	75625.4664	
Coffs Harbour	699.0982	699.0982	
Bikes	699.0982	699.0982	
Road Bikes	699.0982	699.0982	

First Column Stripe

Country	(All)				
Sum of Sales Amount	Column Labels				
	2001				
	3				
	July	July Total	August	August Total	September
Row Labels	No Discount		No Discount		No Discount
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54
Road-150 Red, 48					3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Second Column Stripe

Country	(All)				
Sum of Sales Amount	Column Labels				
	2001				
	3				
	July	July Total	August	August Total	September
Row Labels	No Discount		No Discount		No Discount
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54
Road-150 Red, 48					3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27	
Road-650 Red, 44	699.0982	699.0982			
Darlinghurst	3578.27	3578.27	3578.27	3578.27	7156.54
Bikes	3578.27	3578.27	3578.27	3578.27	7156.54
Road Bikes	3578.27	3578.27	3578.27	3578.27	7156.54
Road-150 Red, 48			3578.27	3578.27	
Road-150 Red, 56	3578.27	3578.27			3578.27
Road-150 Red, 62					3578.27

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

First Row Stripe

Country	(All)			
Sum of Sales Amount	Column Labels			
	2001			
	3			
	July	July Total	August	August Total
Row Labels	No Discount		No Discount	
New South Wales	75625.4664	75625.4664	119823.881	119823.881
Coffs Harbour	699.0982	699.0982	7156.54	7156.54
Bikes	699.0982	699.0982	7156.54	7156.54
Road Bikes	699.0982	699.0982	7156.54	7156.54
Road-150 Red, 48				
Road-150 Red, 52			3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982		
Darlinghurst	3578.27	3578.27	3578.27	3578.27
Bikes	3578.27	3578.27	3578.27	3578.27
Road Bikes	3578.27	3578.27	3578.27	3578.27
Road-150 Red, 48			3578.27	3578.27
Road-150 Red, 56	3578.27	3578.27		
Road-150 Red, 62				

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Second Row Stripe

Country	(All)			
Sum of Sales Amount	Column Labels			
	2001			
	3			
	July	July Total	August	August Total
Row Labels	No Discount		No Discount	
New South Wales	75625.4664	75625.4664	119823.881	119823.881
Coffs Harbour	699.0982	699.0982	7156.54	7156.54
Bikes	699.0982	699.0982	7156.54	7156.54
Road Bikes	699.0982	699.0982	7156.54	7156.54
Road-150 Red, 48				
Road-150 Red, 52			3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982		
Darlinghurst	3578.27	3578.27	3578.27	3578.27
Bikes	3578.27	3578.27	3578.27	3578.27
Road Bikes	3578.27	3578.27	3578.27	3578.27
Road-150 Red, 48			3578.27	3578.27
Road-150 Red, 56	3578.27	3578.27		
Road-150 Red, 62				

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

First Column

Country	(All)		
Sum of Sales Amount	Column Labels		
	2001		
		3	
	July	July Total	August
Row Labels	No Discount		No Discount
New South Wales	75625.4664	75625.4664	119823.881
Coffs Harbour	699.0982	699.0982	7156.54
Bikes	699.0982	699.0982	7156.54
Road Bikes	699.0982	699.0982	7156.54
Road-150 Red, 48			
Road-150 Red, 52			3578.27
Road-150 Red, 62			3578.27
Road-650 Red, 44	699.0982	699.0982	
Darlinghurst	3578.27	3578.27	3578.27
Bikes	3578.27	3578.27	3578.27
Road Bikes	3578.27	3578.27	3578.27
Road-150 Red, 48			3578.27
Road-150 Red, 56	3578.27	3578.27	
Road-150 Red, 62			

Header Row

Country	(All)		
Sum of Sales Amount	Column Labels		
	2001		
		3	
	July	July Total	August
Row Labels	No Discount		No Discount
New South Wales	75625.4664	75625.4664	119823.881
Coffs Harbour	699.0982	699.0982	7156.54
Bikes	699.0982	699.0982	7156.54
Road Bikes	699.0982	699.0982	7156.54
Road-150 Red, 48			
Road-150 Red, 52			3578.27
Road-150 Red, 62			3578.27
Road-650 Red, 44	699.0982	699.0982	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

First Header Cell

Country	(All)		
Sum of Sales Amount	Column Labels		
	2001		
	3		
	July	July Total	August
Row Labels	No Discount		No Discount
New South Wales	75625.4664	75625.4664	119823.881
Coffs Harbour	699.0982	699.0982	7156.54
Bikes	699.0982	699.0982	7156.54
Road Bikes	699.0982	699.0982	7156.54
Road-150 Red, 48			
Road-150 Red, 52			3578.27
Road-150 Red, 62			3578.27
Road-650 Red, 44	699.0982	699.0982	

Subtotal Column 1

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							2001 Total
	3						3 Total	
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road-150 Red, 48					3578.27	3578.27	3578.27	3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27	3578.27	7156.54	7156.54
Road-150 Red, 62			3578.27	3578.27			3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982					699.0982	699.0982

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Subtotal Column 2

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							2001 Total
	3						3 Total	
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road-150 Red, 48					3578.27	3578.27	3578.27	3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27	3578.27	7156.54	7156.54
Road-150 Red, 62			3578.27	3578.27			3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982					699.0982	699.0982

Subtotal Column 3

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							2001 Total
	3						3 Total	
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road-150 Red, 48					3578.27	3578.27	3578.27	3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27	3578.27	7156.54	7156.54
Road-150 Red, 62			3578.27	3578.27			3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982					699.0982	699.0982

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Subtotal Row 1

Country	(All)				
Sum of Sales Amount	Column Labels				
	2001				
	3				
	July	July Total	August	August Total	September
Row Labels	No Discount		No Discount		No Discount
Bendigo					
Mountain Bikes					
Mountain-100 Black, 38	3374.99	3374.99			
Mountain-100 Silver, 42	3399.99	3399.99			
Mountain-100 Silver, 48	3399.99	3399.99			
Mountain Bikes Total	10174.97	10174.97			
Road Bikes					
Road-150 Red, 44	3578.27	3578.27			
Road-150 Red, 52	3578.27	3578.27			
Road-150 Red, 62	3578.27	3578.27			3578.27
Road Bikes Total	10734.81	10734.81			3578.27
Bendigo Total	20909.78	20909.78			3578.27
Brisbane					
Road Bikes					
Road-150 Red, 44	3578.27	3578.27	3578.27	3578.27	
Road-150 Red, 62			3578.27	3578.27	
Road Bikes Total	3578.27	3578.27	7156.54	7156.54	
Brisbane Total	3578.27	3578.27	7156.54	7156.54	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Subtotal Row 2

Country	(All)					
Sum of Sales Amount	Column Labels					
	2001					
		3				
	July	July Total	August	August Total	September	
Row Labels	No Discount		No Discount		No Discount	
Bendigo						
Mountain Bikes						
Mountain-100 Black, 38	3374.99	3374.99				
Mountain-100 Silver, 42	3399.99	3399.99				
Mountain-100 Silver, 48	3399.99	3399.99				
Mountain Bikes Total	10174.97	10174.97				
Road Bikes						
Road-150 Red, 44	3578.27	3578.27				
Road-150 Red, 52	3578.27	3578.27				
Road-150 Red, 62	3578.27	3578.27				3578.27
Road Bikes Total	10734.81	10734.81				3578.27
Bendigo Total	20909.78	20909.78				3578.27
Brisbane						
Road Bikes						
Road-150 Red, 44	3578.27	3578.27	3578.27	3578.27		
Road-150 Red, 62			3578.27	3578.27		
Road Bikes Total	3578.27	3578.27	7156.54	7156.54		

Subtotal Row 3

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Country	(All)					
Sum of Sales Amount	Column Labels					
	2001					
	3					
	July	July Total	August	August Total	September	
Row Labels	No Discount		No Discount		No Discount	
New South Wales						
Coffs Harbour						
Road Bikes						
Road-150 Red, 48						3578.27
Road-150 Red, 52			3578.27	3578.27		3578.27
Road-150 Red, 62			3578.27	3578.27		
Road-650 Red, 44	699.0982	699.0982				
Road Bikes Total	699.0982	699.0982	7156.54	7156.54		7156.54
Coffs Harbour Total	699.0982	699.0982	7156.54	7156.54		7156.54
Darlinghurst						
Road Bikes						
Road-150 Red, 48			3578.27	3578.27		
Road-150 Red, 56	3578.27	3578.27				3578.27
Road-150 Red, 62						3578.27
Road Bikes Total	3578.27	3578.27	3578.27	3578.27		7156.54
Darlinghurst Total	3578.27	3578.27	3578.27	3578.27		7156.54

Column Subheading 1

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							2001 Total
	3						3 Total	
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Column Subheading 2

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							
	3							3 Total
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064		279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54		15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54		15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54		15012.1782

Column Subheading 3

Country	(All)							
Sum of Sales Amount	Column Labels							
	2001							2001 Total
	3						3 Total	
	July	July Total	August	August Total	September	September Total		
Row Labels	No Discount		No Discount		No Discount			
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782

Row Subheading 1

Country	(All)				
Sum of Sales Amount	Column Labels				
	2001				
	3				
	July	July Total	August	August Total	September
Row Labels	No Discount		No Discount		No Discount
New South Wales					
Coffs Harbour					
Road Bikes					
Road-150 Red, 48					3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27	
Road-650 Red, 44	699.0982	699.0982			
Road Bikes Total	699.0982	699.0982	7156.54	7156.54	7156.54
Coffs Harbour Total	699.0982	699.0982	7156.54	7156.54	7156.54

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Row Subheading 2

Country	(All)					
Sum of Sales Amount	Column Labels					
	2001					
		3				
	July		July Total	August	August Total	September
Row Labels	No Discount			No Discount		No Discount
New South Wales						
Coffs Harbour						
Road Bikes						
Road-150 Red, 48						3578.27
Road-150 Red, 52				3578.27	3578.27	3578.27
Road-150 Red, 62				3578.27	3578.27	
Road-650 Red, 44	699.0982		699.0982			
Road Bikes Total	699.0982		699.0982	7156.54	7156.54	7156.54
Coffs Harbour Total	699.0982		699.0982	7156.54	7156.54	7156.54
Darlinghurst						
Road Bikes						
Road-150 Red, 48				3578.27	3578.27	
Road-150 Red, 56	3578.27		3578.27			3578.27
Road-150 Red, 62						3578.27
Road Bikes Total	3578.27		3578.27	3578.27	3578.27	7156.54
Darlinghurst Total	3578.27		3578.27	3578.27	3578.27	7156.54
Goulburn						
Mountain Bikes						
Mountain-100 Silver, 44	3399.99		3399.99			

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Row Subheading 3

Country	(All)				
Sum of Sales Amount	Column Labels				
	2001				
		3			
	July	July Total	August	August Total	September
Row Labels	No Discount		No Discount		No Discount
New South Wales					
Coffs Harbour					
Road Bikes					
Road-150 Red, 48					3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27
Road-150 Red, 62			3578.27	3578.27	
Road-650 Red, 44	699.0982	699.0982			
Road Bikes Total	699.0982	699.0982	7156.54	7156.54	7156.54
Coffs Harbour Total	699.0982	699.0982	7156.54	7156.54	7156.54
Darlinghurst					
Road Bikes					
Road-150 Red, 48			3578.27	3578.27	
Road-150 Red, 56	3578.27	3578.27			3578.27
Road-150 Red, 62					3578.27
Road Bikes Total	3578.27	3578.27	3578.27	3578.27	7156.54
Darlinghurst Total	3578.27	3578.27	3578.27	3578.27	7156.54
Goulburn					
Mountain Bikes					
Mountain-100 Silver, 44	3399.99	3399.99			

Grand Total Column

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Country	(All)								
Sum of Sales Amount	Column Labels								
	2001							2001 Total	Grand Total
	3						3 Total		
	July	July Total	August	August Total	September	September Total			
Row Labels	No Discount		No Discount		No Discount				
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538	279147.7538
Coffs Harbour	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782	15012.1782
Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782	15012.1782
Road Bikes	699.0982	699.0982	7156.54	7156.54	7156.54	7156.54	15012.1782	15012.1782	15012.1782
Road-150 Red, 48					3578.27	3578.27	3578.27	3578.27	3578.27
Road-150 Red, 52			3578.27	3578.27	3578.27	3578.27	7156.54	7156.54	7156.54
Road-150 Red, 62			3578.27	3578.27			3578.27	3578.27	3578.27
Road-650 Red, 44	699.0982	699.0982					699.0982	699.0982	699.0982

Grand Total Row

Country	(All)								
Sum of Sales Amount	Column Labels								
	2001							2001 Total	Grand Total
	3						3 Total		
	July	July Total	August	August Total	September	September Total			
Row Labels	No Discount		No Discount		No Discount				
New South Wales	75625.4664	75625.4664	119823.881	119823.881	83698.4064	83698.4064	279147.7538	279147.7538	279147.7538
Queensland	53228.4682	53228.4682	35376.14	35376.14	29821.0764	29821.0764	118425.6846	118425.6846	118425.6846
South Australia	7156.54	7156.54	11255.6282	11255.6282	6978.26	6978.26	25390.4282	25390.4282	25390.4282
Tasmania	6978.26	6978.26	10531.53	10531.53	3578.27	3578.27	21088.06	21088.06	21088.06
Victoria	66664.17	66664.17	45551.11	45551.11	49917.5	49917.5	162132.78	162132.78	162132.78
Grand Total	209652.9046	209652.9046	222538.2892	222538.2892	173993.5128	173993.5128	606184.7066	606184.7066	606184.7066

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
blankRow (Blank Row Style)	Table style element that applies to PivotTable's blank rows.
firstColumn (First Column Style)	Table style element that applies to table's first column.
firstColumnStripe (First Column Stripe Style)	Table style element that applies to table's first column stripes.
firstColumnSubheading (First Column Subheading Style)	Table style element that applies to PivotTable's first column subheading.
firstHeaderCell (First Header Row Style)	Table style element that applies to table's first header row cell.

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
firstRowStripe (First Row Stripe Style)	Table style element that applies to table's first row stripes.
firstRowSubheading (First Row Subheading Style)	Table style element that applies to PivotTable's first row subheading.
firstSubtotalColumn (First Subtotal Column Style)	Table style element that applies to PivotTable's first subtotal column.
firstSubtotalRow (First Subtotal Row Style)	Table style element that applies to pivot table's first subtotal row.
firstTotalCell (First Total Row Style)	Table style element that applies to table's first total row cell.
headerRow (Header Row Style)	Table style element that applies to table's header row.
lastColumn (Last Column Style)	Table style element that applies to table's last column.
lastHeaderCell (Last Header Style)	Table style element that applies to table's last header row cell.
lastTotalCell (Last Total Row Style)	Table style element that applies to table's last total row cell.
pageFieldLabels (Page Field Labels Style)	Table style element that applies to pivot table's page field labels.
pageFieldValues (Page Field Values Style)	Table style element that applies to pivot table's page field values.
secondColumnStripe (Second Column Stipe Style)	Table style element that applies to table's second column stripes.
secondColumnSubheading (Second Column Subheading Style)	Table style element that applies to pivot table's second column subheading.
secondRowStripe (Second Row Stripe Style)	Table style element that applies to table's second row stripes.
secondRowSubheading (Second Row Subheading Style)	Table style element that applies to pivot table's second row subheading.
secondSubtotalColumn (Second Subtotal Column Style)	Table style element that applies to PivotTable's second subtotal column.
secondSubtotalRow (Second Subtotal Row Style)	Table style element that applies to PivotTable's second subtotal row.
thirdColumnSubheading (Third Column Subheading Style)	Table style element that applies to PivotTable's third column subheading.
thirdRowSubheading (Third Row Subheading Style)	Table style element that applies to PivotTable's third row subheading.
thirdSubtotalColumn (Third Subtotal Column Style)	Table style element that applies to pivot table's third subtotal column.
thirdSubtotalRow (Third Subtotal Row Style)	Table style element that applies to PivotTable's third

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	subtotal row.
totalRow (Total Row Style)	Table style element that applies to table's total row.
wholeTable (Whole Table Style)	Table style element that applies to table's entire content.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TableStyleType) is located in §A.2. *end note*]

18.18.78 ST_TableType (Table Type)

An enumeration that specifies what the table data is based on.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
queryTable (Query Table)	A table based on an external data query.
worksheet (Worksheet)	A table based on a worksheet data range.
xml (XML)	A table based on an XML mapping.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TableType) is located in §A.2. *end note*]

18.18.79 ST_TargetScreenSize (Target Screen Size Types)

This simple type defines the collection of screen resolutions that are supported for this workbook.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1024x768 (1024 x 768 Resolution)	Sets the target screen resolution to 1024x768 pixels.
1152x882 (1152 x 882 Resolution)	Sets the target screen resolution to 1152x882 pixels.
1152x900 (1152 x 900 Resolution)	Sets the target screen resolution to 1152x900 pixels
1280x1024 (1280 x 1024 Resolution)	Sets the target screen resolution to 1280x1024 pixels.
1600x1200 (1600 x 1200 Resolution)	Sets the target screen resolution to 1600x1200 pixels.
1800x1440 (1800 x 1440 Resolution)	Sets the target screen resolution to 1800x1440 pixels.
1920x1200 (1920 x 1200 Resolution)	Sets the target screen resolution to 1920x1200 pixels.
544x376 (544 x 376 Resolution)	Sets the target screen resolution to 544x376 pixels.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
640x480 (640 x 480 Resolution)	Sets the target screen resolution to 640x480 pixels.
720x512 (720 x 512 Resolution)	Sets the target screen resolution to 720x512 pixels.
800x600 (800 x 600 Resolution)	Sets the target screen resolution to 800x600 pixels.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TargetScreenSize) is located in §A.2. *end note*]

18.18.80 ST_TextHAlign (Comment Text Horizontal Alignment)

This simple type specifies the horizontal alignment of the text within a comment text field.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center Alignment)	Specifies that the text is centered horizontally.
distributed (Distributed Alignment)	Specifies that the text is distributed horizontally.
justify (Justify Alignment)	Specifies that the text is justified horizontally.
left (Left Alignment)	Specifies that the text is left-aligned.
right (Right Alignment)	Specifies that the text is right-aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextHAlign) is located in §A.2. *end note*]

18.18.81 ST_TextVAlign (Comment Text Vertical Alignment)

This simple type specifies the vertical alignment of the text within a comment text field.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Bottom Alignment)	Specifies that the text is bottom-aligned.
center (Center Alignment)	Specifies that the text is centered vertically.
distributed (Distributed Alignment)	Specifies that the text is distributed vertically.
justify (Justify Alignment)	Specifies that the text is justified vertically.
top (Top Alignment)	Specifies that the text is top-aligned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextVAlign) is located in §A.2. *end note*]

18.18.82 ST_TimePeriod (Time Period Types)

Used in a "contains dates" conditional formatting rule. These are dynamic time periods, which change based on the date the conditional formatting is refreshed / applied.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
last7Days (Last 7 Days)	A date in the last seven days.
lastMonth (Last Month)	A date occuring in the last calendar month.
lastWeek (Last Week)	A date occuring last week.
nextMonth (Next Month)	A date occuring in the next calendar month.
nextWeek (Next Week)	A date occuring next week.
thisMonth (This Month)	A date occuring in this calendar month.
thisWeek (This Week)	A date occuring this week.
today (Today)	Today's date.
tomorrow (Tomorrow)	Tomorrow's date.
yesterday (Yesterday)	Yesterday's date.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TimePeriod) is located in §A.2. *end note*]

18.18.83 ST_TotalsRowFunction (Totals Row Function Types)

An enumeration that specifies what function is used to aggregate the data in a column before it is displayed in the totals row.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
average (Average)	Represents the arithmetic mean.
count (Non Empty Cell Count)	Represents a count of the number of non-empty cells.
countNums (Count Numbers)	Represents the number of cells that contain numbers.
custom (Custom Formula)	Represents the formula provided in `totalsRowFormula`.
max (Maximum)	Represents the largest value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
min (Minimum)	Represents the smallest value.
none (None)	No total row.
stdDev (StdDev)	Represents the estimated standard deviation.
sum (Sum)	Represents the arithmetic sum.
var (Var)	Represents the estimated variance.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TotalsRowFunction) is located in §A.2. *end note*]

18.18.84 ST_Type (Top N Evaluation Type)

This simple type defines the values for the Top N conditional formatting evaluation for the PivotTable. For more information on Top N conditional formatting, see the Sheet (§18.3.1) reference material.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Indicates that Top N conditional formatting is evaluated across the entire scope range.
column (Column Top N)	Indicates that Top N conditional formatting is evaluated for each column.
none (Top N None)	Indicates that Top N conditional formatting is not evaluated
row (Row Top N)	Indicates that Top N conditional formatting is evaluated for each row.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Type) is located in §A.2. *end note*]

18.18.85 ST_UnderlineValues (Underline Types)

Represents the different types of possible underline formatting.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
double (Double Underline)	Double-line underlining under each character in the cell. underlines are drawn through the descenders of characters such as g and p.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	1.23
doubleAccounting (Accounting Double Underline)	Double-line accounting underlining under each character in the cell. The underlines are drawn under the descenders of characters such as g and p. 1.23
none (None)	No underline.
single (Single Underline)	Single-line underlining under each character in the cell. The underline is drawn through the descenders of characters such as g and p. 1.23
singleAccounting (Accounting Single Underline)	Single-line accounting underlining under each character in the cell. The underline is drawn under the descenders of characters such as g and p. 1.23

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnderlineValues) is located in §A.2. *end note*]

18.18.86 ST_UnsignedIntHex (Hex Unsigned Integer)

This simple type represents the Hex representation of an unsigned integer.

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 8 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedIntHex) is located in §A.2. *end note*]

18.18.87 ST_UpdateLinks (Update Links Behavior Types)

This simple type defines when the application updates links to other workbooks when the workbook is opened.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
always (Always Update Links)	Indicates that links to other workbooks are always updated when the workbook is opened. The application will not display an alert in the user interface (UI).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
never (Never Update Links)	Indicates that links to other workbooks are never updated when the workbook is opened. The application will not display an alert in the user interface.
userSet (User Set)	Indicates that the end-user specified whether they receive an alert to update links to other workbooks when the workbook is opened. [*Example*: The application can expose this option in an application settings dialog. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UpdateLinks) is located in §A.2. *end note*]

18.18.88 ST_VerticalAlignment (Vertical Alignment Types)

This enumeration value indicates the type of vertical alignment for a cell, i.e., whether it is aligned top, bottom, vertically centered, justified or distributed.

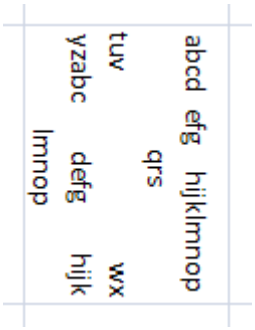
This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Aligned To Bottom)	The vertical alignment is aligned-to-bottom.
center (Centered Vertical Alignment)	The vertical alignment is centered across the height of the cell.
distributed (Distributed Vertical Alignment)	When text direction is horizontal: the vertical alignment of lines of text is distributed vertically, where each line of text inside the cell is evenly distributed across the height of the cell, with flush top and bottom margins. When text direction is vertical: behaves exactly as distributed horizontal alignment. The first words in a line of text (appearing at the top of the cell) are flush with the top edge of the cell, and the last words of a line of text are flush with the bottom edge of the cell, and the line of text is distributed evenly from top to bottom. [*Example*: Horizontal text: this first example shows four lines of text (read horizontally from left to right) distributed vertically across the height of the cell. The first line is "abc", the second line is "def", the third line

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

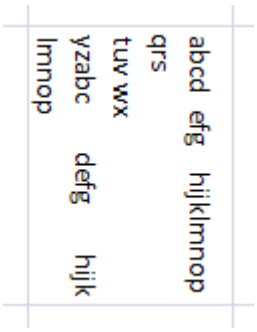
Enumeration Value	Description
	is "ghi" and the fourth line is "jkl". Vertical text: this second example shows three lines of text (read vertically from top to bottom) distributed vertically across the height of the cell. The lines of text are: abcd efg hijklmnop qrs tuv wx yzabc defg hijk lmnop The rendering looks like this:  *end example*]
justify (Justified Vertically)	When text direction is horizontal: the vertical alignment of lines of text is distributed vertically,

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	where each line of text inside the cell is evenly distributed across the height of the cell, with flush top and bottom margins. When text direction is vertical: similar behavior as horizontal justification. The alignment is justified (flush top and bottom in this case). For each line of text, each line of the wrapped text in a cell is aligned to the top and bottom (except the last line). If no single line of text wraps in the cell, then the text is not justified. [*Example*: Horizontal text: this first example shows four lines of text (read horizontally from left to right) justified vertically across the height of the cell. The first line is "abc", the second line is "def", the third line is "ghi" and the fourth line is "jkl". Vertical text: this second example shows three lines of text (read vertically from top to bottom) distributed vertically across the height of the cell. The lines of text are: abcd efg hijklmnop qrs tuv wx yzabc defg hijk lmnop The rendering looks like this:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	 *end example*]
top (Align Top)	The vertical alignment is aligned-to-top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAlignment) is located in §A.2. *end note*]

18.18.89 ST_Visibility (Visibility Types)

This simple type defines the possible states for workbook window visibility.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hidden (Hidden)	Indicates the workbook window is hidden, but can be shown by the user via the user interface.
veryHidden (Very Hidden)	Indicates the workbook window is hidden and cannot be shown in the user interface (UI). This state is only available programmatically.
visible (Visible)	Indicates the workbook window is visible.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Visibility) is located in §A.2. *end note*]

18.18.90 ST_VolDepType (Volatile Dependency Types)

This simple type defines the dependency types available for this workbook.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
olapFunctions (OLAP Formulas)	Indicates that the type is Cube Functions.
realTimeData (Real Time Data)	Indicates that the type is Real Time Data (RTD).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VolDepType) is located in §A.2. *end note*]

18.18.91 ST_VolValueType (Volatile Dependency Value Types)

This simple type defines the data type of the values in the dependency cache.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Boolean)	Indicates topic value is a boolean.
e (Error)	Indicates topic value is an error.
n (Real Number)	Indicates topic value is a real number.
s (String)	Indicates topic value is a string.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VolValueType) is located in §A.2. *end note*]

18.18.92 ST_WebSourceType (Web Source Type)

This is an enumeration of types of objects which can be selected from the workbook to be published as HTML. For example, the entire sheet can be published, or a narrower set of objects on the sheet can be published, like a chart or a range.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autoFilter (AutoFilter)	Auto filter
chart (Chart)	Chart
label (Label)	Label
pivotTable (PivotTable)	PivotTable
printArea (Print Area)	Print area
query (QueryTable)	Query Table

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
range (Range)	Range of cells
sheet (All Sheet Content)	All content of a sheet

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WebSourceType) is located in §A.2. *end note*]

18.18.93 ST_XmlDataType (XML Data Types)

Represents a W3C XML built-in datatype name (http://www.w3.org/TR/xmlschema-2/). The values permitted by this type are the names of the simple datatypes defined by the XMLSchema Library, http://www.w3.org/2001/XML-Schema-datatypes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

Note: The W3C XML Schema definition of this simple type's content model (ST_XmlDataType) is located in §A.2. *end note*]

18.18.94 ST_FontFamily (Font Family)

This simple type specifies a font family. A font family is a set of fonts having common stroke width and serif characteristics. This is system-level font information.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type is restricted to the values listed in the following table:

Value	Font Family
0	Not applicable.
1	Roman
2	Swiss
3	Modern
4	Script
5	Decorative
6	Reserved for future use
7	Reserved for future use
8	Reserved for future use
9	Reserved for future use
10	Reserved for future use
11	Reserved for future use
12	Reserved for future use
13	Reserved for future use

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Value	Font Family
14	Reserved for future use

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontFamily) is located in §A.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19. PresentationML Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, “Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)”. *end note*]

The subordinate subclauses specify the semantics for the XML markup comprising a PresentationML document, as defined by §13 of this Part of ISO/IEC 29500.

19.1 Table of Contents

This subclause is informative.

19.2 Presentation 2522
19.2.1 Presentation Properties 2522
19.2.1.1 bold (Bold Embedded Font) 2522
19.2.1.2 boldItalic (Bold Italic Embedded Font) 2523
19.2.1.3 browse (Browse Slide Show Mode) 2524
19.2.1.4 clrMru (Color MRU) 2524
19.2.1.5 custShow (Custom Show) 2525
19.2.1.6 custShow (Custom Show) 2526
19.2.1.7 custShowLst (List of Custom Shows) 2526
19.2.1.8 defaultTextStyle (Presentation Default Text Style) 2526
19.2.1.9 embeddedFont (Embedded Font) 2527
19.2.1.10 embeddedFontLst (Embedded Font List) 2527
19.2.1.11 ext (Extension) 2527
19.2.1.12 extLst (Extension List) 2528
19.2.1.13 font (Embedded Font Name) 2528
19.2.1.14 handoutMasterId (Handout Master ID) 2531
19.2.1.15 handoutMasterIdLst (List of Handout Master IDs) 2532
19.2.1.16 italic (Italic Embedded Font) 2532
19.2.1.17 kinsoku (Kinsoku Settings) 2533
19.2.1.18 kiosk (Kiosk Slide Show Mode) 2533
19.2.1.19 modifyVerifier (Modification Verifier) 2534
19.2.1.20 notesMasterId (Notes Master ID) 2537
19.2.1.21 notesMasterIdLst (List of Notes Master IDs) 2538
19.2.1.22 notesSz (Notes Slide Size) 2538
19.2.1.23 penClr (Pen Color for Slide Show) 2539
19.2.1.24 photoAlbum (Photo Album Information) 2539
19.2.1.25 present (Presenter Slide Show Mode) 2540
19.2.1.26 presentation (Presentation) 2541
19.2.1.27 presentationPr (Presentation-wide Properties) 2543
19.2.1.28 prnPr (Printing Properties) 2543
19.2.1.29 regular (Regular Embedded Font) 2544
19.2.1.30 showPr (Presentation-wide Show Properties) 2545

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.2.1.31 sld (Presentation Slide) 2545
19.2.1.32 sldAll (All Slides) 2546
19.2.1.33 sldId (Slide ID) 2546
19.2.1.34 sldIdLst (List of Slide IDs) 2547
19.2.1.35 sldLst (List of Presentation Slides) 2547
19.2.1.36 sldMasterId (Slide Master ID) 2547
19.2.1.37 sldMasterIdLst (List of Slide Master IDs) 2548
19.2.1.38 sldRg (Slide Range) 2548
19.2.1.39 sldSz (Presentation Slide Size) 2549
19.2.1.40 smartTags (Smart Tags) 2550
19.2.2 View Properties 2551
19.2.2.1 cSldViewPr (Common Slide View Properties) 2551
19.2.2.2 cViewPr (Common View Properties) 2551
19.2.2.3 gridSpacing (Grid Spacing) 2552
19.2.2.4 guide (A Guide) 2553
19.2.2.5 guideLst (List of Guides) 2553
19.2.2.6 normalViewPr (Normal View Properties) 2553
19.2.2.7 notesTextViewPr (Notes Text View Properties) 2555
19.2.2.8 notesViewPr (Notes View Properties) 2555
19.2.2.9 origin (View Origin) 2555
19.2.2.10 outlineViewPr (Outline View Properties) 2556
19.2.2.11 restoredLeft (Normal View Restored Left Properties) 2556
19.2.2.12 restoredTop (Normal View Restored Top Properties) 2556
19.2.2.13 scale (View Scale) 2557
19.2.2.14 sld (Presentation Slide) 2557
19.2.2.15 sldLst (List of Presentation Slides) 2558
19.2.2.16 slideViewPr (Slide View Properties) 2558
19.2.2.17 sorterViewPr (Slide Sorter View Properties) 2558
19.2.2.18 viewPr (Presentation-wide View Properties) 2559

19.3 Slides 2559

19.3.1 Slides 2559
19.3.1.1 bg (Slide Background) 2559
19.3.1.2 bgPr (Background Properties) 2560
19.3.1.3 bgRef (Background Style Reference) 2560
19.3.1.4 blipFill (Picture Fill) 2561
19.3.1.5 bodyStyle (Slide Master Body Text Style) 2564
19.3.1.6 clrMap (Color Scheme Map) 2564
19.3.1.7 clrMapOvr (Color Scheme Map Override) 2566
19.3.1.8 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties) 2566
19.3.1.9 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties) 2566
19.3.1.10 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties) 2567
19.3.1.11 cNvPicPr (Non-Visual Picture Drawing Properties) 2567
19.3.1.12 cNvPr (Non-Visual Drawing Properties) 2568
19.3.1.13 cNvSpPr (Non-Visual Drawing Properties for a Shape) 2570
19.3.1.14 contentPart (Content Part) 2571
19.3.1.15 controls (List of controls) 2574
19.3.1.16 cSld (Common Slide Data) 2574

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.3.1.17 custData (Customer Data) 2575
19.3.1.18 custDataLst (Customer Data List) 2575
19.3.1.19 cxnSp (Connection Shape) 2575
19.3.1.20 extLst (Extension List with Modification Flag) 2576
19.3.1.21 graphicFrame (Graphic Frame) 2577
19.3.1.22 grpSp (Group Shape) 2577
19.3.1.23 grpSpPr (Group Shape Properties) 2578
19.3.1.24 handoutMaster (Handout Master) 2579
19.3.1.25 hf (Header/Footer information for a slide master) 2579
19.3.1.26 notes (Notes Slide) 2580
19.3.1.27 notesMaster (Notes Master) 2580
19.3.1.28 notesStyle (Notes Text Style) 2581
19.3.1.29 nvCxnSpPr (Non-Visual Properties for a Connection Shape) 2581
19.3.1.30 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame) 2581
19.3.1.31 nvGrpSpPr (Non-Visual Properties for a Group Shape) 2581
19.3.1.32 nvPicPr (Non-Visual Properties for a Picture) 2581
19.3.1.33 nvPr (Non-Visual Properties) 2582
19.3.1.34 nvSpPr (Non-Visual Properties for a Shape) 2582
19.3.1.35 otherStyle (Slide Master Other Text Style) 2582
19.3.1.36 ph (Placeholder Shape) 2583
19.3.1.37 pic (Picture) 2583
19.3.1.38 sld (Presentation Slide) 2584
19.3.1.39 sldLayout (Slide Layout) 2585
19.3.1.40 sldLayoutId (Slide Layout Id) 2586
19.3.1.41 sldLayoutIdLst (List of Slide Layouts) 2586
19.3.1.42 sldMaster (Slide Master) 2586
19.3.1.43 sp (Shape) 2587
19.3.1.44 spPr (Shape Properties) 2587
19.3.1.45 spTree (Shape Tree) 2588
19.3.1.46 style (Shape Style) 2589
19.3.1.47 tags (Customer Data Tags) 2590
19.3.1.48 timing (Slide Timing Information for a Slide Layout) 2590
19.3.1.49 titleStyle (Slide Master Title Text Style) 2591
19.3.1.50 transition (Slide Transition for a Slide Layout) 2591
19.3.1.51 txBody (Shape Text Body) 2592
19.3.1.52 txStyles (Slide Master Text Styles) 2592
19.3.1.53 xfrm (2D Transform for Graphic Frame) 2592
19.3.2 Embedded Objects 2593
19.3.2.1 control (Embedded Control) 2594
19.3.2.2 embed (Embedded Object or Control) 2594
19.3.2.3 link (Linked Object or Control) 2595
19.3.2.4 oleObj (Global Element for Embedded objects and Controls) 2595
19.3.3 Programmable Tags 2596
19.3.3.1 tag (Programmable Extensibility Tag) 2596
19.3.3.2 tagLst (Programmable Tab List) 2596
19.4 Comments 2597
19.4.1 cm (Comment) 2597

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.4.2 cmAuthor (Comment Author) 2598
19.4.3 cmAuthorLst (List of Comment Authors) 2599
19.4.4 cmLst (Comment List) 2599
19.4.5 pos (Comment Position) 2600
19.4.6 text (Comment's Text Content) 2601

19.5 Animation 2601

19.5.1 anim (Animate) 2602
19.5.2 animClr (Animate Color Behavior) 2603
19.5.3 animEffect (Animate Effect) 2604
19.5.4 animMotion (Animate Motion) 2610
19.5.5 animRot (Animate Rotation) 2611
19.5.6 animScale (Animate Scale) 2612
19.5.7 attrName (Attribute Name) 2613
19.5.8 attrNameLst (Attribute Name List) 2614
19.5.9 audio (Audio) 2615
19.5.10 bg (Background) 2616
19.5.11 bldAsOne (Build As One) 2616
19.5.12 bldDgm (Build Diagram) 2616
19.5.13 bldGraphic (Build Graphics) 2617
19.5.14 bldLst (Build List) 2618
19.5.15 bldOleChart (Build Embedded Chart) 2619
19.5.16 bldP (Build Paragraph) 2620
19.5.17 bldSub (Build Sub Elements) 2622
19.5.18 blinds (Blinds Slide Transition) 2622
19.5.19 boolVal (Boolean Variant) 2623
19.5.20 by (By) 2624
19.5.21 by (By) 2624
19.5.22 cBhvr (Common Behavior) 2625
19.5.23 charRg (Character Range) 2626
19.5.24 checker (Checker Slide Transition) 2627
19.5.25 childTnLst (Children Time Node List) 2628
19.5.26 circle (Circle Slide Transition) 2629
19.5.27 clrVal (Color Value) 2629
19.5.28 cmd (Command) 2630
19.5.29 cMediaNode (Common Media Node Properties) 2632
19.5.30 comb (Comb Slide Transition) 2633
19.5.31 cond (Condition) 2633
19.5.32 cover (Cover Slide Transition) 2634
19.5.33 cTn (Common Time Node Properties) 2636
19.5.34 cut (Cut Slide Transition) 2639
19.5.35 diamond (Diamond Slide Transition) 2640
19.5.36 dissolve (Dissolve Slide Transition) 2640
19.5.37 endCondLst (End Conditions List) 2641
19.5.38 endSnd (Stop Sound Action) 2642
19.5.39 endSync (EndSync) 2642
19.5.40 excl (Exclusive) 2643
19.5.41 fade (Fade Slide Transition) 2643

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.42 fltVal (Float Value) 2644
19.5.43 from (From) 2645
19.5.44 from (From) 2645
19.5.45 graphicEl (Graphic Element) 2646
19.5.46 hsl (HSL) 2647
19.5.47 inkTgt (Ink Target) 2647
19.5.48 intVal (Integer) 2648
19.5.49 iterate (Iterate) 2648
19.5.50 newsflash (Newsflash Slide Transition) 2649
19.5.51 nextCondLst (Next Conditions List) 2650
19.5.52 oleChartEl (Embedded Chart Element) 2650
19.5.53 par (Parallel Time Node) 2651
19.5.54 plus (Plus Slide Transition) 2652
19.5.55 prevCondLst (Previous Conditions List) 2653
19.5.56 pRg (Paragraph Text Range) 2653
19.5.57 progress (Progress) 2654
19.5.58 pull (Pull Slide Transition) 2654
19.5.59 push (Push Slide Transition) 2656
19.5.60 random (Random Slide Transition) 2658
19.5.61 randomBar (Random Bar Slide Transition) 2658
19.5.62 rCtr (Rotation Center) 2659
19.5.63 rgb (RGB) 2660
19.5.64 rtn (Runtime Node Trigger Choice) 2661
19.5.65 seq (Sequence Time Node) 2661
19.5.66 set (Set Time Node Behavior) 2663
19.5.67 sldTgt (Slide Target) 2664
19.5.68 snd (Sound) 2664
19.5.69 sndAc (Sound Action) 2665
19.5.70 sndTgt (Sound Target) 2666
19.5.71 split (Split Slide Transition) 2666
19.5.72 spTgt (Shape Target) 2668
19.5.73 stCondLst (Start Conditions List) 2668
19.5.74 strips (Strips Slide Transition) 2669
19.5.75 strVal (String Value) 2670
19.5.76 stSnd (Start Sound Action) 2671
19.5.77 subSp (Subshape) 2671
19.5.78 subTnLst (Sub-TimeNodes List) 2672
19.5.79 tav (Time Animate Value) 2672
19.5.80 tavLst (Time Animated Value List) 2677
19.5.81 tgtEl (Target Element) 2678
19.5.82 tmAbs (Time Absolute) 2678
19.5.83 tmPct (Time Percentage) 2679
19.5.84 tmpl (Template Effects) 2679
19.5.85 tmplLst (Template effects) 2680
19.5.86 tn (Time Node) 2680
19.5.87 tnLst (Time Node List) 2681
19.5.88 to (To) 2682
19.5.89 to (To) 2682

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.5.90 to (To) 2683
19.5.91 txEl (Text Element) 2683
19.5.92 val (Value) 2684
19.5.93 video (Video) 2684
19.5.94 wedge (Wedge Slide Transition) 2685
19.5.95 wheel (Wheel Slide Transition) 2686
19.5.96 wipe (Wipe Slide Transition) 2688
19.5.97 zoom (Zoom Slide Transition) 2689

19.6 Slide Synchronization Data 2690
19.6.1 sldSyncPr (Slide Synchronization Properties) 2690

19.7 Simple Types 2691
19.7.1 ST_BookmarkIdSeed (Bookmark ID Seed) 2691
19.7.2 ST_Direction (Direction) 2691
19.7.3 ST_Index (Index) 2691
19.7.4 ST_IterateType (Iterate Type) 2692
19.7.5 ST_Name (Name string) 2692
19.7.6 ST_OleObjectFollowColorScheme (Embedded object to Follow Color Scheme) 2692
19.7.7 ST_PhotoAlbumFrameShape (Photo Album Shape for Photo Mask) 2693
19.7.8 ST_PhotoAlbumLayout (Photo Album Layout Definition) 2694
19.7.9 ST_PlaceholderSize (Placeholder Size) 2695
19.7.10 ST_PlaceholderType (Placeholder IDs) 2696
19.7.11 ST_PrintColorMode (Print Color Mode) 2697
19.7.12 ST_PrintWhat (Default print output) 2697
19.7.13 ST_SlideId (Slide Identifier) 2698
19.7.14 ST_SlideLayoutId (Slide Layout ID) 2698
19.7.15 ST_SlideLayoutType (Slide Layout Type) 2698
19.7.16 ST_SlideMasterId (Slide Master ID) 2702
19.7.17 ST_SlideSizeCoordinate (Slide Size Coordinate) 2702
19.7.18 ST_SlideSizeType (Slide Size Type) 2702
19.7.19 ST_SplitterBarState (Splitter Bar State) 2703
19.7.20 ST_TLAnimateBehaviorCalcMode (Time List Animate Behavior Calculate Mode) 2703
19.7.21 ST_TLAnimateBehaviorValueType (Time List Animate Behavior Value Types) 2704
19.7.22 ST_TLAnimateColorDirection (Time List Animate Color Direction) 2704
19.7.23 ST_TLAnimateColorSpace (Time List Animate Color Space) 2704
19.7.24 ST_TLAnimateEffectTransition (Time List Animate Effect Transition) 2705
19.7.25 ST_TLAnimateMotionBehaviorOrigin (Time List Animate Motion Behavior Origin) 2705
19.7.26 ST_TLAnimateMotionPathEditMode (Time List Animate Motion Path Edit Mode) 2705
19.7.27 ST_TLBehaviorAccumulateType (Behavior Accumulate Type) 2706
19.7.28 ST_TLBehaviorAdditiveType (Behavior Additive Type) 2706
19.7.29 ST_TLBehaviorOverrideType (Behavior Override Type) 2707
19.7.30 ST_TLBehaviorTransformType (Behavior Transform Type) 2707
19.7.31 ST_TLChartSubelementType (Chart Subelement Type) 2707
19.7.32 ST_TLCommandType (Command Type) 2708
19.7.33 ST_TLDiagramBuildType (Diagram Build Types) 2708
19.7.34 ST_TLNextActionType (Next Action Type) 2709
19.7.35 ST_TLOleChartBuildType (Embedded Chart Build Type) 2709
19.7.36 ST_TLParaBuildType (Paragraph Build Type) 2710

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.7.37 ST_TLPreviousActionType (Previous Action Type) 2710
19.7.38 ST_TLTime (Time) 2711
19.7.39 ST_TLTimeAnimateValueTime (Animation Time) 2711
19.7.40 ST_TLTimeIndefinite (Indefinite Time Declaration) 2711
19.7.41 ST_TLTimeNodeFillType (Time Node Fill Type) 2711
19.7.42 ST_TLTimeNodeID (Time Node ID) 2712
19.7.43 ST_TLTimeNodeMasterRelation (Time Node Master Relation) 2712
19.7.44 ST_TLTimeNodePresetClassType (Time Node Preset Class Type) 2712
19.7.45 ST_TLTimeNodeRestartType (Time Node Restart Type) 2713
19.7.46 ST_TLTimeNodeSyncType (Time Node Sync Type) 2713
19.7.47 ST_TLTimeNodeType (Time Node Type) 2714
19.7.48 ST_TLTriggerEvent (Trigger Event) 2714
19.7.49 ST_TLTriggerRuntimeNode (Trigger RunTime Node) 2715
19.7.50 ST_TransitionCornerDirectionType (Transition Corner Direction Type) 2715
19.7.51 ST_TransitionEightDirectionType (Transition Eight Direction) 2716
19.7.52 ST_TransitionInOutDirectionType (Transition In/Out Direction Type) 2716
19.7.53 ST_TransitionSideDirectionType (Transition Side Direction Type) 2716
19.7.54 ST_TransitionSpeed (Transition Speed) 2717
19.7.55 ST_ViewType (List of View Types) 2717

End of informative text.

19.2 Presentation

The Presentation portion of the PresentationML framework houses a set of elements that describe the storing of presentation-wide and view-specific properties. The presentation-wide properties are those that pertain to the entire presentation. The view-specific properties assist the generating application and viewing application by storing parameters that pertain to the final delivery of the presentation.

19.2.1 Presentation Properties

This section contains all presentation-level properties that pertain to a presentation document:

19.2.1.1 bold (Bold Embedded Font)

This element specifies a bold embedded font that is linked to a parent typeface. Once specified, this bold version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example*: Consider the following embedded font with a bold version specified.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:bold r:id="rId2"/>
</p:embeddedFont>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontDataId) is located in §A.3. *end note*]

19.2.1.2 boldItalic (Bold Italic Embedded Font)

This element specifies a bold italic embedded font that is linked to a parent typeface. Once specified, this bold italic version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example*: Consider the following embedded font with a bold italic version specified.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:boldItalic r:id="rId2"/>
</p:embeddedFont>
```

end example]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontDataId) is located in §A.3. *end note*]

19.2.1.3 browse (Browse Slide Show Mode)

This element specifies that the presentation slide show should be viewed in a single window or browse mode, instead of full screen.

[*Example*: Consider the following presentation that is to be viewed in a browse mode.

```
<p:presentationPr xmlns:a="…" xmlns:r="…" xmlns:p="…">
  <p:showPr>
    …
    <p:browse showScrollbar="0"/>
    …
  </p:showPr>
</p:presentationPr>
```

end example]

Attributes	Description
showScrollbar (Show Scroll Bar in Window)	Specifies whether to show the scroll bar in the viewing window. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShowInfoBrowse) is located in §A.3. *end note*]

19.2.1.4 clrMru (Color MRU)

This specifies the most recently used user-selected colors within the presentation. This list contains custom user-selected colors outside the presentation's theme colors, enabling the application to expose these additional color choices for easy reuse. The first item in the list is the most recently used color.

[*Example*: Consider the following presentation with two user-selected colors in the color MRU list.

```
<p:presentationPr xmlns:a="…" xmlns:r="…" xmlns:p="…">
  …
  <p:clrMru>
    <a:srgbClr val="5361EB"/>
    <a:srgbClr val="CCECFF"/>
  </p:clrMru>
  …
</p:presentationPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMRU) is located in §A.4.1. *end note*]

19.2.1.5 custShow (Custom Show)

This element specifies a custom show which is an ordered list of a group of slides that are contained within the presentation. The custom show element allows for the specification of a presentation order that is different from the order in which the slides themselves are stored.

[*Example*: Consider the following custom show list that outlines a couple custom shows for a given set of slides.

```
<p:custShowLst>
  <p:custShow name="Custom Show 1" id="0">
    <p:sldLst>
      <p:sld r:id="rId4"/>
      <p:sld r:id="rId3"/>
      <p:sld r:id="rId2"/>
      <p:sld r:id="rId5"/>
    </p:sldLst>
  </p:custShow>
  <p:custShow name="Custom Show 2" id="1">
    <p:sldLst>
      <p:sld r:id="rId4"/>
      <p:sld r:id="rId5"/>
    </p:sldLst>
  </p:custShow>
</p:custShowLst>
```

In the above example there are two custom shows specified. The first specifies to present the slides in the order of 4, 3, 2 then 5 while the second specifies to play only slide 4 then 5. *end example*]

Attributes	Description
id (Custom Show Identifier)	This attribute specifies the custom show identification number. This is a number given that should be unique within the presentation document. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomShowId) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.2.1.6 custShow (Custom Show)

This element specifies a custom show that defines a specific slide sequence that the slides are displayed in. This allows for many variants of the same set of slides to be presented.

[*Example*: Consider the following custom show using three slides.

```
<p:custShow name="Custom Show 1" id="0">
  <p:sldLst>
    <p:sld r:id="rId5"/>
    <p:sld r:id="rId2"/>
    <p:sld r:id="rId4"/>
  </p:sldLst>
</p:custShow>
```

Notice here that the custom show specifies a show, or presentation, where slide 5 is shown first, then slide 2 and finally slide 4. *end example*]

Attributes	Description
id (Custom Show ID)	Specifies the identification number for this custom show. This should be unique among all the custom shows within the corresponding presentation. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
name (Custom Show Name)	Specifies a name for the custom show. The possible values for this attribute are defined by the ST_Name simple type (§19.7.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomShow) is located in §A.3. *end note*]

19.2.1.7 custShowLst (List of Custom Shows)

This element specifies a list of all custom shows that are available within the corresponding presentation. A custom show is a defined slide sequence that allows for the displaying of the slides with the presentation in any arbitrary order.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomShowList) is located in §A.3. *end note*]

19.2.1.8 defaultTextStyle (Presentation Default Text Style)

This element specifies the default text styles that are to be used within the presentation. The text style defined here can be referenced when inserting a new slide if that slide is not associated with a master slide or if no styling information has been otherwise specified for the text within the presentation slide.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

19.2.1.9 embeddedFont (Embedded Font)

This element specifies an embedded font. Once specified, this font is available for use within the presentation. Within a font specification there can be regular, bold, italic and boldItalic versions of the font specified. The actual font data for each of these is referenced using a relationships file that contains links to all available fonts. This font data contains font information for each of the characters to be made available in each version of the font.

[*Example*: Consider the following embedded font.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:regular r:id="rId2"/>
</p:embeddedFont>
```

end example]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontListEntry) is located in §A.3. *end note*]

19.2.1.10 embeddedFontLst (Embedded Font List)

This element specifies a list of fonts that are embedded within the corresponding presentation. The font data for these fonts is stored alongside the other document parts within the document container. The actual font data is referenced within the embeddedFont element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontList) is located in §A.3. *end note*]

19.2.1.11 ext (Extension)

This element specifies an extension that is used for future extensions to the current version of DrawingML. This allows for the specifying of currently unknown elements for later versions of generating applications.

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
uri (Uniform Resource Identifier)	This attribute specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of this tag.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Extension) is located in §A.3. *end note*]

19.2.1.12 extLst (Extension List)

This element specifies the extension list within which all future extensions of element type ext are defined. The extension list along with corresponding future extensions is used to extend the storage capabilities of the PresentationML framework. This allows for various new kinds of data to be stored natively within the framework.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExtensionList) is located in §A.3. *end note*]

19.2.1.13 font (Embedded Font Name)

This element specifies specific properties describing an embedded font. Once specified, this font is available for use within the presentation. Within a font specification there can be regular, bold, italic and boldItalic versions of the font specified. The actual font data for each of these is referenced using a relationships file that contains links to all available fonts. This font data contains font information for each of the characters to be made available in each version of the font.

[*Example*: Consider the following embedded font:

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:regular r:id="rId2"/>
</p:embeddedFont>
```

end example]

Font Substitution Logic:

If the specified font is not available on a system being used for rendering, then the attributes of this element are to be utilized in selecting an alternate font.

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

If ambiguities exist between properties in the markup of a body of text that refers to an embedded font (such as the latin element specified in §21.1.2.3.7) and properties in the markup of the corresponding instance of this embedded font element, the determination whether to use that embedded font is application-dependent behavior. If ambiguities exist between properties in the markup of an instance of this embedded font element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

and properties within the corresponding embedded Font part as specified in §15.2.13, the determination whether to use that embedded font is application-dependent behavior.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set that is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

The value of this attribute shall be interpreted as follows:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set.</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>(IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>
The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>panose (Panose Setting)</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §4.2.7.17 of ISO/IEC</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	14496-22:2007. The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41).
typeface (Text Typeface) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

19.2.1.14 handoutMasterId (Handout Master ID)

This element specifies a handout master that is available within the corresponding presentation. A handout master is a slide that is specifically designed for printing as a handout.

[*Example*: Consider the following specification of a handout master within a presentation

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:handoutMasterIdLst>
    <p:handoutMasterId r:id="rId8"/>
  </p:handoutMasterIdLst>
  …
</p:presentation>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the handoutMaster element defining this handout master. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HandoutMasterIdListEntry) is located in §A.3. *end note*]

19.2.1.15 handoutMasterIdLst (List of Handout Master IDs)

This element specifies a list of identification information for the handout master slides that are available within the corresponding presentation. A handout master is a slide that is specifically designed for printing as a handout.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HandoutMasterIdList) is located in §A.3. *end note*]

19.2.1.16 italic (Italic Embedded Font)

This element specifies an italic embedded font that is linked to a parent typeface. Once specified, this italic version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example*: Consider the following embedded font with a italic version specified.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:italic r:id="rId2"/>
</p:embeddedFont>
```

end example]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Attributes	Description
id (Relationship Identifier) Namespace:	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
http://purl.oclc.org/ooxml/officeDocument/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontDataId) is located in §A.3. *end note*]

19.2.1.17 kinsoku (Kinsoku Settings)

This element specifies the presentation-wide kinsoku settings that define the line breaking behaviour of East Asian text within the corresponding presentation.

Attributes	Description
invalEndChars (Invalid Kinsoku End Characters)	Specifies the characters that cannot end a line of text. The possible values for this attribute are defined by the W3C XML Schema string datatype.
invalStChars (Invalid Kinsoku Start Characters)	Specifies the characters that cannot start a line of text. The possible values for this attribute are defined by the W3C XML Schema string datatype.
lang (Language)	Specifies the corresponding East Asian language that these settings apply to. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Kinsoku) is located in §A.3. *end note*]

19.2.1.18 kiosk (Kiosk Slide Show Mode)

This element specifies that the presentation slide show should be viewed in a full-screen kiosk mode. A presentation viewed in kiosk mode should have user input disabled and restarts after a specified interval.

[*Example*: Consider the following presentation that is set to be viewed in a looping kiosk mode.

```
<p:presentationPr xmlns:a="…" xmlns:r="…" xmlns:p="…">
  <p:showPr loop="1" showNarration="1">
    …
    <p:kiosk/>
    …
  </p:showPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</p:presentationPr>
```

end example]

Attributes	Description
restart (Restart Show)	Specifies the time length that the presentation should run until it is to be restarted. That is, the presentation should loop back to the first slide specified in the presentation or custom show. This value is specified in 1/1000ths of a second and measured from the most recent time the presentation started or restarted. [*Note*: The counter is reset when a presentation is restarted due to automatic looping at the end of a show, if specified by the loop attribute of showPr. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShowInfoKiosk) is located in §A.3. *end note*]

19.2.1.19 modifyVerifier (Modification Verifier)

This element specifies the write protection settings which have been applied to a PresentationML document. *Write protection* refers to a mode in which the document's contents should not be modified, and the document should not be resaved using the same file name.

When present, the application shall require a password to enable modifications to the document. If the supplied password does not match the hash value in this attribute, then write protection shall be enabled. If this element is omitted, then no write protection shall be applied to the current document. Since this protection does not encrypt the document, malicious applications might circumvent its use.

The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 510 octets are truncated to 510 octets. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation. The attributes of this element specify the algorithm to be used to verify the password provided by the user.

[*Example*: Consider a PresentationML document that can only be opened in a write protected state unless a password is provided, in which case the file would be opened in an editable state. This requirement would be specified using the following PresentationML:

```
<p:modifyVerifier p:algorithmName="SHA-512" …
p:hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" … />
```

…In order for the hosting application to enable edits to the document, the hosting application would have to be provided with a password that the hosting application would then hash using the algorithm specified by the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

algorithm attributes and compare to the value of the hashValue attribute (`9oN7nWkCAyEZib1RomSJTjmPpCY=`). If the two values matched, the file would be opened in an editable state. *end example*]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>algorithmName (Cryptographic Algorithm Name)</td><td>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and input password in order to compute the hash value.

The following values are reserved:
<table>
<tr><th>Value</th><th>Algorithm</th></tr>
<tr><td>MD2</td><td>Specifies that the MD2 algorithm, as defined by RFC 1319, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD4</td><td>Specifies that the MD4 algorithm, as defined by RFC 1320, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>MD5</td><td>Specifies that the MD5 algorithm, as defined by RFC 1321, shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-128</td><td>Specifies that the RIPEMD-128 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.

[Note: It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks. end note]</td></tr>
<tr><td>RIPEMD-160</td><td>Specifies that the RIPEMD-160 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-1</td><td>Specifies that the SHA-1 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-256</td><td>Specifies that the SHA-256 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-384</td><td>Specifies that the SHA-384 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
<tr><td>SHA-512</td><td>Specifies that the SHA-512 algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td>WHIRLPOOL</td><td>Specifies that the WHIRLPOOL algorithm, as defined by ISO/IEC 10118-3:2004 shall be used.</td></tr></table> [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `< … algorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The algorithmName attribute value of "SHA-1" specifies that the SHA-1 hashing algorithm must be used to generate a hash from the user-defined password. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hashValue (Password Hash Value)	Specifies the hash value for the password required to edit this chartsheet. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced. If this value is omitted, then the reservationPassword attribute shall contain the password hash for the workbook. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… algorithmName="SHA-1" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The hashValue attribute value of 9oN7nWkCAyEZib1RomSJTjmPpCY= specifies that the user-supplied password must be hashed using the pre-processing defined by the parent element (if any) followed by the SHA-1 algorithm (specified via the algorithmName attribute value of SHA-1) and that the resulting has value must be 9oN7nWkCAyEZib1RomSJTjmPpCY= for the protection to be disabled. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
saltValue (Salt Value for Password Verifier)	Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hashValue attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A *salt* is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those pre-calculated values (often referred to as a "dictionary attack").

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value. [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… saltValue="ZUdHa+D8F/OAKP3I7ssUnQ==" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The saltValue attribute value of `ZUdHa+D8F/OAKP3I7ssUnQ==` specifies that the user-supplied password must have this value prepended before it is run through the specified hashing algorithm to generate a resulting hash value for comparison. *end example*] The possible values for this attribute are defined by the W3C XML Schema base64Binary datatype.
spinValue (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (runs using each iteration's result plus a 4 byte value (0-based, little endian) containing the number of the iteration as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hashValue attribute. [*Rationale*: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). *end rationale*] [*Example*: Consider an Office Open XML document with the following information stored in one of its protection elements: `<… spinCount="100000" hashValue="9oN7nWkCAyEZib1RomSJTjmPpCY=" />` The spinCount attribute value of `100000` specifies that the hashing function must be run one hundred thousand times to generate a hash value for comparison with the hashValue attribute. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ModifyVerifier) is located in §A.3. *end note*]

19.2.1.20 notesMasterId (Notes Master ID)

This element specifies a notes master that is available within the corresponding presentation. A notes master is a slide that is specifically designed for the printing of the slide along with any attached notes.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following specification of a notes master within a presentation

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:notesMasterIdLst>
    <p:notesMasterId r:id="rId8"/>
  </p:notesMasterIdLst>
  …
</p:presentation>
```

end example]

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the notesMaster element defining this notes master. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesMasterIdListEntry) is located in §A.3. *end note*]

19.2.1.21 notesMasterIdLst (List of Notes Master IDs)

This element specifies a list of identification information for the notes master slides that are available within the corresponding presentation. A notes master is a slide that is specifically designed for the printing of the slide along with any attached notes.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesMasterIdList) is located in §A.3. *end note*]

19.2.1.22 notesSz (Notes Slide Size)

This element specifies the size of slide surface used for notes slides and handout slides. Objects within a notes slide can be specified outside these extents, but the notes slide has a background surface of the specified size when presented or printed. This element is intended to specify the region to which content is fitted in any special format of printout the application might choose to generate, such as an outline handout.

[*Example*: Consider the following specifying of the size of a notes slide.

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:notesSz cx="9144000" cy="6858000"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    …
</p:presentation>
```

end example]

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

19.2.1.23 penClr (Pen Color for Slide Show)

This element specifies the pen color that should be used to make markings on the slides while in a presentation.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

19.2.1.24 photoAlbum (Photo Album Information)

This element specifies that the corresponding presentation contains a photo album. A photo album specifies a list of images within the presentation that spread across one or more slides, all of which share a consistent layout. Each image in the album is formatted with a consistent style. This functionality enables the application to manage all of the images together and modify their ordering, layout, and formatting as a set.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This element does not enforce the specified properties on individual photo album images; rather, it specifies common settings that should be applied by default to all photo album images and their containing slides. Images that are part of the photo album are identified by the presence of the isPhoto element in the definition of the picture.

[*Example*: Consider the following presentation that has been specified as a photo album

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:photoAlbum bw="1" layout="2pic"/>
  …
</p:presentation>
```

end example]

Attributes	Description
bw (Black and White)	Specifies whether all pictures in the photo album are to be displayed as black and white. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
frame (Frame Type)	Specifies the frame type that is to be used on all the pictures in the photo album. The possible values for this attribute are defined by the ST_PhotoAlbumFrameShape simple type (§19.7.7).
layout (Photo Album Layout)	Specifies the layout that is to be used to arrange the pictures in the photo album on individual slides. The possible values for this attribute are defined by the ST_PhotoAlbumLayout simple type (§19.7.8).
showCaptions (Show/Hide Captions)	Specifies whether to show captions for pictures in the photo album. Captions are text boxes grouped with each image, with the group set to not allow ungrouping. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PhotoAlbum) is located in §A.3. *end note*]

19.2.1.25 present (Presenter Slide Show Mode)

This element specifies that the presentation slide show should be viewed in a full-screen presenter mode. In this mode, the presentation is displayed on one monitor while a different monitor displays notes and provides navigation controls intended to be viewed only by the presenter.

[*Example*: Consider the following presentation that is set to be viewed in a present mode.

 ©ISO/IEC 2016 – All rights reserved

```
<p:presentationPr xmlns:a="…" xmlns:r="…" xmlns:p="…">
  <p:showPr>
    …
    <p:present/>
    …
  </p:showPr>
</p:presentationPr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.2.1.26 presentation (Presentation)

This element specifies within it fundamental presentation-wide properties.

[*Example*: Consider the following presentation with a single slide master and two slides. In addition to these commonly used elements there can also be the specification of other properties such as slide size, notes size and default text styles.

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…">
    <p:sldMasterIdLst>
        <p:sldMasterId id="2147483648" r:id="rId1"/>
    </p:sldMasterIdLst>
    <p:sldIdLst>
        <p:sldId id="256" r:id="rId3"/>
        <p:sldId id="257" r:id="rId4"/>
    </p:sldIdLst>
    <p:sldSz cx="9144000" cy="6858000" type="screen4x3"/>
    <p:notesSz cx="6858000" cy="9144000"/>
    <p:defaultTextStyle>
      …
    </p:defaultTextStyle>
</p:presentation>
```

end example]

Attributes	Description
autoCompressPictures (Automatically Compress Pictures)	Specifies whether the generating application should automatically compress all pictures for this presentation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bookmarkIdSeed (Bookmark ID Seed)	Specifies a seed for generating bookmark IDs to ensure IDs remain unique across the document. This value specifies the number to be used as the ID for the next new bookmark created.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the ST_BookmarkIdSeed simple type (§19.7.1).
compatMode (Compatibility Mode)	Specifies whether the generating application is to be in a compatibility mode which serves to inform the user of any loss of content or functionality when working with older formats. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

conformance (Document Conformance Class)	Specifies the conformance class (§2.1) to which the PresentationML document conforms.
	
	If this attribute is omitted, its default value is `transitional`.
	
	[*Example*: Consider the following PresentationML Presentation part markup:
	
	`<p:presentation conformance="strict">`
	`…`
	`</p:presentation>`
	
	This document has a conformance attribute value of `strict`, therefore it conforms to the PML Strict conformance class. *end example*]
	
	The possible values for this attribute are defined by the ST_ConformanceClass simple type (§22.9.2.2).

embedTrueTypeFonts (Embed True Type Fonts)	Specifies whether the generating application should automatically embed true type fonts or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
firstSlideNum (First Slide Number)	Specifies the first slide number in the presentation. The possible values for this attribute are defined by the W3C XML Schema int datatype.
removePersonalInfoOnSave (Remove Personal Information on Save)	Specifies whether to automatically remove personal information when the presentation document is saved. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rtl (Right-To-Left Views)	Specifies if the current view of the user interface is oriented right-to-left or left-to-right. The view is right-to-left is this value is set to true, and left-to-right otherwise. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
saveSubsetFonts (Save Subset Fonts)	Specifies to save only the subset of characters used in the presentation when a font is embedded.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
serverZoom (Server Zoom)	Specifies the scaling to be used when the presentation is embedded in another document. The embedded slides are to be scaled by this percentage. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
showSpecialPlsOnTitleSld (Show Header and Footer Placeholders on Titles)	Specifies whether to show the header and footer placeholders on the title slides. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
strictFirstAndLastChars (Strict First and Last Characters)	Specifies whether to use strict characters for starting and ending lines of Japanese text. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Presentation) is located in §A.3. *end note*]

19.2.1.27 presentationPr (Presentation-wide Properties)

This element functions as a parent element within which additional presentation-wide document properties are contained. All properties and their corresponding settings are defined within the child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresentationProperties) is located in §A.3. *end note*]

19.2.1.28 prnPr (Printing Properties)

This element specifies the default printing properties associated with this presentation document.

Attributes	Description
clrMode (Print Color Mode)	Specifies the color mode to be used when printing. The possible values for this attribute are defined by the ST_PrintColorMode simple type (§19.7.11).
frameSlides (Frame slides when printing)	Specifies whether slides should be framed when printing. When framed, an outline border is printed for each slide. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hiddenSlides (Print	Specifies whether hidden slides should be printed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Hidden Slides)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
prnWhat (Print Output)	Specifies what the default print output is in terms of content layout. The possible values for this attribute are defined by the ST_PrintWhat simple type (§19.7.12).
scaleToFitPaper (Scale to Fit Paper when printing)	Specifies whether the print output should be scaled to fit the paper being used. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PrintProperties) is located in §A.3. *end note*]

19.2.1.29 regular (Regular Embedded Font)

This element specifies a regular embedded font that is linked to a parent typeface. Once specified, this regular version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example*: Consider the following embedded font with a regular version specified.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:regular r:id="rId2"/>
</p:embeddedFont>
```

end example]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedFontDataId) is located in §A.3. *end note*]

19.2.1.30 showPr (Presentation-wide Show Properties)

This element functions as a parent element within which all presentation-wide show properties are contained. All properties and their corresponding settings are defined within the child elements.

Attributes	Description
loop (Loop Slide Show)	Specifies whether the slide show should be set to loop at the end. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showAnimation (Show Animation in Slide Show)	Specifies whether slide show animation should be shown when presenting. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showNarration (Show Narration in Slide Show)	Specifies whether slide show narration should be played when presenting. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
useTimings (Use Timings in Slide Show)	Specifies whether slide transition timings should be used to advance slides when presenting. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShowProperties) is located in §A.3. *end note*]

19.2.1.31 sld (Presentation Slide)

This element specifies a slide within a slide list. The slide list is used to specify an ordering of slides.

[*Example*: Consider the following custom show with an ordering of slides.

```
<p:custShowLst>
  <p:custShow name="Custom Show 1" id="0">
    <p:sldLst>
      <p:sld r:id="rId4"/>
      <p:sld r:id="rId3"/>
      <p:sld r:id="rId2"/>
      <p:sld r:id="rId5"/>
    </p:sldLst>
  </p:custShow>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
</p:custShowLst>
```

In the above example the order specified to present the slides is slide 4, then 3, 2 and finally 5. *end example*]

Attributes	Description
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This attribute specifies the relationship id that is used to reference to the actual slide XML file that contains all the information to the slide listed within the slide list. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideRelationshipListEntry) is located in §A.3. *end note*]

19.2.1.32 sldAll (All Slides)

This attribute specifies all slides instead of a given range of slides for use within the html publishing properties as well as the show properties.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.2.1.33 sldId (Slide ID)

This element specifies a presentation slide that is available within the corresponding presentation. A slide contains the information that is specific to a single slide such as slide-specific shape and text information.

[*Example*: Consider the following specification of a slide master within a presentation

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  ..
  <p:sldIdLst>
    <p:sldId id="256" r:id="rId3"/>
    <p:sldId id="257" r:id="rId4"/>
    <p:sldId id="258" r:id="rId5"/>
    <p:sldId id="259" r:id="rId6"/>
    <p:sldId id="260" r:id="rId7"/>
  </p:sldIdLst>
  ..
</p:presentation>
```

end example]

Attributes	Description
id (Relationship	Specifies the relationship identifier that is used in conjunction with a corresponding

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	relationship file to resolve the location within a presentation of the sld element defining this slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
id (Slide Identifier)	Specifies the slide identifier that is to contain a value that is unique throughout the presentation. The possible values for this attribute are defined by the ST_SlideId simple type (§19.7.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideIdListEntry) is located in §A.3. *end note*]

19.2.1.34 sldIdLst (List of Slide IDs)

This element specifies a list of identification information for the slides that are available within the corresponding presentation. A slide contains the information that is specific to a single slide such as slide-specific shape and text information.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideIdList) is located in §A.3. *end note*]

19.2.1.35 sldLst (List of Presentation Slides)

This element specifies a list of presentation slides. A presentation slide contains the information that is specific to a single slide such as slide-specific shape and text information.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideRelationshipList) is located in §A.3. *end note*]

19.2.1.36 sldMasterId (Slide Master ID)

This element specifies a slide master that is available within the corresponding presentation. A slide master is a slide that is specifically designed to be a template for all related child layout slides.

[*Example*: Consider the following specification of a slide master within a presentation

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:sldMasterIdLst>
    <p:sldMasterId id="2147483648" r:id="rId1"/>
  </p:sldMasterIdLst>
  …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</p:presentation>
```

end example]

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the sldMaster element defining this slide master. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
id (Slide Master Identifier)	Specifies the slide master identifier that is to contain a value that is unique throughout the presentation. The possible values for this attribute are defined by the ST_SlideMasterId simple type (§19.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideMasterIdListEntry) is located in §A.3. *end note*]

19.2.1.37 sldMasterIdLst (List of Slide Master IDs)

This element specifies a list of identification information for the slide master slides that are available within the corresponding presentation. A slide master is a slide that is specifically designed to be a template for all related child layout slides.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideMasterIdList) is located in §A.3. *end note*]

19.2.1.38 sldRg (Slide Range)

This element specifies a slide range for use within the html publishing properties as well as the show properties.

[*Note*: The indexes used here correlate directly with the presentation slide numbers which they reference to. That is the slide range must be greater than or equal to 1 and also less than or equal to the number of slides in the presentation document. *end note*]

Attributes	Description
end (End)	This attribute defines the end of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).
st (Start)	This attribute defines the start of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_IndexRange) is located in §A.3. *end note*]

19.2.1.39 sldSz (Presentation Slide Size)

This element specifies the size of the presentation slide surface. Objects within a presentation slide can be specified outside these extents, but this is the size of background surface that is shown when the slide is presented or printed..

[*Example*: Consider the following specifying of the size of a presentation slide.

```
<p:presentation xmlns:a="…" xmlns:r="…" xmlns:p="…" embedTrueTypeFonts="1">
  …
  <p:sldSz cx="9144000" cy="6858000" type="screen4x3"/>
  …
</p:presentation>
```

end example]

Attributes	Description
cx (Extent Length)	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_SlideSizeCoordinate simple type (§19.7.17).
cy (Extent Width)	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_SlideSizeCoordinate simple type (§19.7.17).
type (Type of Size)	Specifies the kind of slide size that should be used. This identifies in particular the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	expected delivery platform for this presentation. The possible values for this attribute are defined by the ST_SlideSizeType simple type (§19.7.18).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideSize) is located in §A.3. *end note*]

19.2.1.40 smartTags (Smart Tags)

This element specifies that references to smart tags exist within this document. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*] To denote the location of smart tags on individual runs of text, there smart tag identifier attributes are specified for each run to which a smart tag applies. These are further specified in the run property attributes within DrawingML.

[*Example*: Consider the following PresentationML markup:

```
<p:presentation>
  …
  <p:smartTags r:id="rId1"/>
</p:presentation>
```

The presence of the smartTags element specifies that there is smart tag information within the PresentationML package. Individual runs are then inspected for the value of the smtId attribute to determine where smart tags might apply, for example:

```
<p:txBody>
  <a:bodyPr/>
  <a:lstStyle/>
  <a:p>
    <a:r>
      <a:rPr lang="en-US" dirty="0" smtId="1"/>
      <a:t>CNTS</a:t>
    </a:r>
    <a:endParaRPr lang="en-US" dirty="0"/>
  </a:p>
</p:txBody>
```

In the sample above there is a smart tag identifier of 1 specified for this run of text to denote that the text should be inspected for smart tag information. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this smart tag. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SmartTags) is located in §A.3. *end note*]

19.2.2 View Properties

This section contains all properties that pertain to the viewing of the presentation.

19.2.2.1 cSldViewPr (Common Slide View Properties)

This element functions as a container for slide view properties that are common across multiple view property elements. The specific properties and associated values for these view properties reside within the child elements and attributes.

Attributes	Description
showGuides (Show Guides in View)	Specifies whether to show guides when editing the presentation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
snapToGrid (Snap Objects to Grid)	Specifies whether objects should snap to underlying presentation grid when editing. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
snapToObjects (Snap Objects to Objects)	Specifies whether objects should snap to other objects when editing the presentation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommonSlideViewProperties) is located in §A.3. *end note*]

19.2.2.2 cViewPr (Common View Properties)

This element specifies the view properties that are common across multiple view property elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
varScale (Variable Scale)	Specifies that the view content should automatically scale to best fit the current window size. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommonViewProperties) is located in §A.3. *end note*]

19.2.2.3 gridSpacing (Grid Spacing)

This element specifies the grid spacing that should be used for the grid underlying the presentation document. The grid can be used to align objects on the slide and to display visual positioning cues.

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.2.2.4 guide (A Guide)

This element specifies a guide within the presentation. Guides are lines used for arranging layouts and content and never appear except as an aid in editing slides.

Attributes	Description
orient (Guide Orientation)	Specifies the orientation for a guide. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).
pos (Guide Position)	Specifies the position information for a guide. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Guide) is located in §A.3. *end note*]

19.2.2.5 guideLst (List of Guides)

This element specifies a list of guides for a particular view of the presentation.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GuideList) is located in §A.3. *end note*]

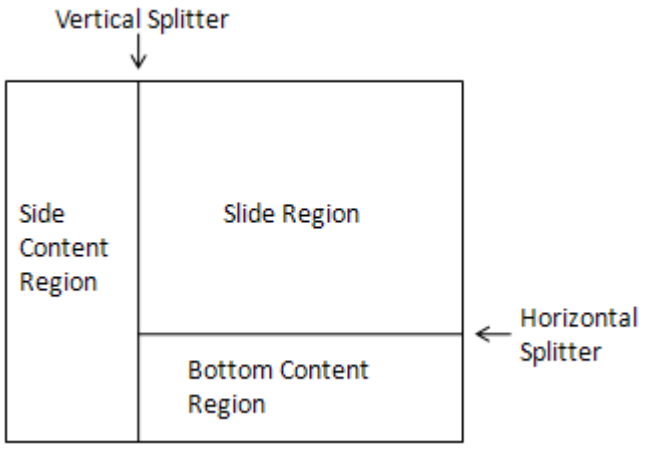
19.2.2.6 normalViewPr (Normal View Properties)

This element specifies the view properties associated with the normal slide view (§19.7.55) mode. The normal slide view consists of three content regions: the slide itself, a side content region, and a bottom content region. The content of the side content region and bottom content region is determined by the generating application. Properties pertaining to the positioning of the different content regions are stored in this element. This information allows the application to save its view state to the file, so that when reopened the view is in the same state as when the presentation was last saved.

A vertical splitter bar separates the slide from the side content region. A horizontal splitter bar separates the slide from the content region below the slide. If the presentation is set to left-to-right, the side content region is to the left of the slide. If the presentation is set to right-to-left, the side content region is to the right of the slide.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



Attributes	Description
horzBarState (State of the Horizontal Splitter Bar)	Specifies the state that the horizontal splitter bar should be in when in normal slide view mode (§19.7.55). The region to be maximized or minimized is the side content region. The possible values for this attribute are defined by the ST_SplitterBarState simple type (§19.7.19).
preferSingleView (Prefer Single View)	Specifies whether the user prefers to see a full-window single-content region over the normal slide view(§19.7.55) with three content regions. If enabled, the application can choose to display one of the content regions in the entire window. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showOutlineIcons (Show Outline Icons in Normal View)	Specifies whether the application should show icons if displaying outline content in any of the content regions of normal slide view mode (§19.7.55). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
snapVertSplitter (Snap Vertical Splitter)	Specifies whether the vertical splitter should snap to a minimized state when the side region is sufficiently small. The specific parameters of this behaviour are left to the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vertBarState (State of the Vertical Splitter Bar)	Specifies the state that the vertical splitter bar should be in when in normal slide view mode (§19.7.55). The region to be maximized or minimized is the primary slide content region. The possible values for this attribute are defined by the ST_SplitterBarState simple type (§19.7.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NormalViewProperties) is located in §A.3. *end note*]

19.2.2.7 notesTextViewPr (Notes Text View Properties)

This element functions as a parent element within which all properties associated with the notes text view are contained. All properties are defined within the child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesTextViewProperties) is located in §A.3. *end note*]

19.2.2.8 notesViewPr (Notes View Properties)

This element functions as a parent element within which all view properties associated with notes are contained. All properties are defined within the child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesViewProperties) is located in §A.3. *end note*]

19.2.2.9 origin (View Origin)

This element specifies the origin of the slide when it is being viewed with various scaling factors using the scale element.

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

19.2.2.10 outlineViewPr (Outline View Properties)

This element functions as a parent element within which all view properties associated with the outline view (§19.7.55) mode are contained. All properties are defined within the child elements.

Outline view displays only the textual content of a presentation. The presentation is formatted as an outline, with slide titles as the first level of the outline. Body text on slides is indented below the slide title.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OutlineViewProperties) is located in §A.3. *end note*]

19.2.2.11 restoredLeft (Normal View Restored Left Properties)

This element specifies the sizing of the side content region of the normal slide view (§19.7.55), when the region is of a variable restored size (neither minimized nor maximized).

Attributes	Description
autoAdjust (Auto Adjust Normal View)	Specifies whether the size of the side content region should compensate for the new size when resizing the window containing the view within the application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sz (Normal View Dimension Size)	Specifies the size of the slide region (width when a child of restoredTop, height when a child of restoredLeft). The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NormalViewPortion) is located in §A.3. *end note*]

19.2.2.12 restoredTop (Normal View Restored Top Properties)

This element specifies the sizing of the top slide region of the normal slide view (§19.7.55), when the region is of a variable restored size (neither minimized nor maximized).

Attributes	Description
autoAdjust (Auto Adjust Normal	Specifies whether the size of the side content region should compensate for the new size when resizing the window containing the view within the application.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
View)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sz (Normal View Dimension Size)	Specifies the size of the slide region (width when a child of restoredTop, height when a child of restoredLeft). The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_NormalViewPortion) is located in §A.3. *end note*]

19.2.2.13 scale (View Scale)

This element specifies the view scaling factors that the presentation was last viewed with.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scale2D) is located in §A.4.1. *end note*]

19.2.2.14 sld (Presentation Slide)

This element specifies a presentation slide and properties specific to the slide's appearance in outline view (§19.7.55).

[*Example*: Consider the following presentation slide that has been collapsed in outline view.

```
<p:viewPr xmlns:a="…" xmlns:r="…" xmlns:p="…" lastView="outlineView">
  …
  <p:outlineViewPr>
    …
    <p:sldLst>
      <p:sld r:id="rId1" collapse="1"/>
    </p:sldLst>
    …
  </p:outlineViewPr>
  …
</p:viewPr>
```

end example]

Attributes	Description
collapse (Collapsed)	Specifies whether this presentation slide is to be shown as collapsed within outline view. That is, all text other than the slide title is not shown to the user.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Relationship Identifier) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this presentation slide within a presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OutlineViewSlideEntry) is located in §A.3. *end note*]

19.2.2.15 sldLst (List of Presentation Slides)

This element specifies a list of presentation slides. A presentation slide contains the information that is specific to a single slide such as slide-specific shape and text information.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OutlineViewSlideList) is located in §A.3. *end note*]

19.2.2.16 slideViewPr (Slide View Properties)

This element functions as a parent element within which all view properties associated with the slide view mode are contained. All properties are defined within the child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideViewProperties) is located in §A.3. *end note*]

19.2.2.17 sorterViewPr (Slide Sorter View Properties)

This element functions as a parent element within which all view properties associated with the slide sorter view (§19.7.55) mode are contained. All properties are defined within the child elements.

The slide sorter view displays thumbnails of multiple slides at once; the number of slides and size of thumbnails depends on the scaling factor of the view.

Attributes	Description
showFormatting (Show Formatting)	Specifies whether to show associated slide formatting when in slide sorter view mode. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideSorterViewProperties) is located in §A.3. *end note*]

19.2.2.18 viewPr (Presentation-wide View Properties)

This element functions as a parent element within which all presentation-wide view properties are contained. All properties and their corresponding settings are defined within the child elements.

Attributes	Description
lastView (Last View)	Specifies the view mode that was used when the presentation document was last saved. The possible values for this attribute are defined by the ST_ViewType simple type (§19.7.55).
showComments (Show Comments)	Specifies whether the slide comments should be shown. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ViewProperties) is located in §A.3. *end note*]

19.3 Slides

The Slides portion of the PresentationML framework stores all information pertaining specifically to slides of various slide types. These slide types and corresponding parts can be broken down into three distinct parts, namely slides, embedded objects, and programmable tags.

19.3.1 Slides

Being the main segment of this section of PresentationML, the slides elements encompass all data that is to be contained within a slide. The best way to think of a slide is a container for all data that is to be on that slide. The specific shapes, images and relations within a slide do not come into play here. The elements here pertain to the six different slide types that can be described within PresentationML, namely slide, slide layout, slide master, handout master, notes master and notes slide.

19.3.1.1 bg (Slide Background)

This element specifies the background appearance information for a slide. The slide background covers the entire slide and is visible where no objects exist and as the background for transparent objects.

Attributes	Description
bwMode (Black and White Mode)	Specifies that the background should be rendered using only black and white coloring. That is, the coloring information for the background should be converted to either black or white when rendering the picture. [*Note*: No gray is to be used in rendering this background, only stark black and stark

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	white. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Background) is located in §A.3. *end note*]

19.3.1.2 bgPr (Background Properties)

This element specifies visual effects used to render the slide background. This includes any fill, image, or effects that are to make up the background of the slide.

Attributes	Description
shadeToTitle (Shade to Title)	Specifies whether the background of the slide is of a shade to title background type. This kind of gradient fill is on the slide background and changes based on the placement of the slide title placeholder. An example is shown below. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BackgroundProperties) is located in §A.3. *end note*]

19.3.1.3 bgRef (Background Style Reference)

This element specifies the slide background is to use a fill style defined in the style matrix. The idx attribute refers to the index of a background fill style or fill style within the presentation's style matrix, defined by the fmtScheme element. A value of 0 or 1000 indicates no background, values 1-999 refer to the index of a fill style within the fillStyleLst element, and values 1001 and above refer to the index of a background fill style within

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

the bgFillStyleLst element. The value 1001 corresponds to the first background fill style, 1002 to the second background fill style, and so on.

[*Example*:

```
<p:bgRef idx="2">
  <a:schemeClr val="bg2"/>
</p:bgRef>
```

The above code indicates a slide background with the style's second fill style using the second background color of the color scheme.

end example]

[*Example*:

```
<p:bgRef idx="1001">
  <a:schemeClr val="bg2"/>
</p:bgRef>
```

The above code indicates a slide background with the style's first background fill style using the second background color of the color scheme.

end example]

Attributes	Description
idx (Style Matrix Index) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

19.3.1.4 blipFill (Picture Fill)

This element specifies the kind of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<p:pic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    <p:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </p:blipFill>
    …
  </p:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
  <p:pic>
    …
    <p:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </p:blipFill>
    <p:spPr>
      <a:gradFill>
        <a:gsLst>
          <a:gs pos="0">
            <a:schemeClr val="tx2">
              <a:shade val="50000"/>
            </a:schemeClr>
          </a:gs>
          <a:gs pos="39999">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <a:schemeClr val="tx2">
              <a:tint val="20000"/>
            </a:schemeClr>
          </a:gs>
          <a:gs pos="70000">
            <a:srgbClr val="C4D6EB"/>
          </a:gs>
          <a:gs pos="100000">
            <a:schemeClr val="bg1"/>
          </a:gs>
        </a:gsLst>
      </a:gradFill>
    </p:spPr>
    …
  </p:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace:	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
http://purl.oclc.org/ooxml/drawingml/main	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

19.3.1.5 bodyStyle (Slide Master Body Text Style)

This element specifies the text formatting style for all body text within a master slide. This formatting is used on all body text within presentation slides related to this master. The text formatting is specified by utilizing the DrawingML framework just as within a regular presentation slide. Within the bodyStyle element there can be many different style types defined as there are different kinds of text stored within the body of a slide.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

19.3.1.6 clrMap (Color Scheme Map)

This element specifies the mapping layer that transforms one color scheme definition to another. Each attribute represents a color name that can be referenced in this master, and the value is the corresponding color in the theme.

[*Example*: Consider the following mapping of colors that applies to a slide master:

```
<p:clrMap bg1="dk1" tx1="lt1" bg2="dk2" tx2="lt2" accent1="accent1"
accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
accent6="accent6" hlink="hlink" folHlink="folHlink"/>
```

end example]

Attributes	Description
accent1 (Accent 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent2 (Accent 2) Namespace: http://purl.oclc.org/ooxml/drawing	Specifies a color defined which is associated as the accent 2 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
ml/main	
accent3 (Accent 3) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent4 (Accent 4) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 4 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent5 (Accent 5) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent6 (Accent 6) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg1 (Background 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	A color defined which is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg2 (Background 2) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the second background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
folHlink (Followed Hyperlink) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
hlink (Hyperlink) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx1 (Text 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx2 (Text 2) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

19.3.1.7 clrMapOvr (Color Scheme Map Override)

This element provides a mechanism with which to override the color schemes listed within the ClrMap element. If the masterClrMapping element is present, the color scheme defined by the master is used. If the overrideClrMapping element is present, it defines a new color scheme specific to the parent notes slide, presentation slide, or slide layout.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMappingOverride) is located in §A.4.1. *end note*]

19.3.1.8 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual drawing properties specific to a connector shape. This includes information specifying the shapes to which the connector shape is connected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

19.3.1.9 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

19.3.1.10 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

19.3.1.11 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

19.3.1.12 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: <… descr="A picture of a bowl of fruit"> The descr attribute contains alternative text which can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… title="Process Flow Diagram"> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

19.3.1.13 cNvSpPr (Non-Visual Drawing Properties for a Shape)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with

[*Example*: Consider the shape that has a shape lock applied to it.

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="2" name="Rectangle 1"/>
    <p:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </p:cNvSpPr>
   </p:nvSpPr>
   …
</p:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.3.1.14 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ISO/IEC 29500. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[*Note*: For better interoperability, only standard XML formats should be used. *end note*]

[*Example*: Consider a PresentationML document which includes the following SMIL markup in a part named smil1.xml:

```
<!--
            Copyright: Copyright 1998-2001 W3C (MIT, INRIA, Keio), All Rights
Reserved.
See http://www.w3.org/Consortium/Legal/.
Author: Aaron Cohen (Intel)
           Version: February 7, 2001
            Module: Animation Module
           Feature: animation
         File Name: animation-add-BE-05.smil
  Media Components: none
 Expected Behavior: Nine red rectangles numbered 1 to 9 shrink to squares
                    over 2s as follows:
                    at 2s #1 shrinks.
                    at 5s #2 shrinks, 1s after #1 completes
                    at 8s #3 shrinks.
                    #4 shrinks when it is clicked on.
                    #5 shrinks 1s after it is clicked on.
                    #6 shrinks 2s after it is clicked on.
                    #7 shrinks when the accesskey '1' is pressed.
                    #8 should be shrunk from 0s since it's wallclock time is in
the past.
                    #9 will not shrink unless a DOM call causes it to begin.
-->
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout width="640" height="480" backgroundColor="white"/>
      <region id="whole" width="640" height="480" z-index="0"/>
      <region id="rect1" top="50px" left="90px" height="50px" width="30px"
backgroundColor="red" z-index="1"/>
      <region id="rect2" top="50px" left="234px" height="50px" width="30px"
backgroundColor="red" z-index="1"/>
      <region id="rect4" top="160px" left="90px" height="50px" width="30px"
backgroundColor="transparent" z-index="1"/>
      <region id="rect5" top="160px" left="234px" height="50px" width="30px"
backgroundColor="transparent" z-index="1"/>
      <region id="rect6" top="160px" left="380px" height="50px" width="30px"
backgroundColor="transparent" z-index="1"/>
      <region id="rect7" top="270px" left="90px" height="50px" width="30px"
backgroundColor="red" z-index="1"/>
      <region id="rect8" top="270px" left="234px" height="50px" width="30px"
backgroundColor="red" z-index="1"/>
      <region id="rect9" top="270px" left="380px" height="50px" width="30px"
backgroundColor="red" z-index="1"/>
    </layout>
  </head>
<!-- Copyright 1998-2001 W3C (MIT, INRIA, Keio), All Rights Reserved.
See http://www.w3.org/Consortium/Legal/. -->
  <body>
    <par dur="indefinite">
      <img src="../images/animation-timing-BE-05.jpg" region="whole"/>
      <animate id="anim1" targetElement="rect1" attributeName="height" from="50"
to="25" begin="2s" dur="2s" fill="freeze"/>
      <animate id="anim2" targetElement="rect2" attributeName="height" from="50"
to="25" begin="anim1.end+1s" dur="2s" fill="freeze"/>
      <brush id="brush4" color="red" region="rect4" height="50px" width="30px"/>
      <animate id="anim4" targetElement="brush4" attributeName="height"
from="50" to="25" begin="brush4.activateEvent" dur="2s" fill="freeze"/>
      <brush id="brush5" color="red" region="rect5" height="50px" width="30px"/>
      <animate id="anim5" targetElement="brush5" attributeName="height"
from="50" to="25" begin="brush5.activateEvent+1s" dur="2s" fill="freeze"/>
      <brush id="brush6" color="red" region="rect6" height="50px" width="30px"/>
      <animate id="anim6a" targetElement="brush6" attributeName="width"
repeatCount="3" from="30" to="30" begin="brush6.activateEvent" dur="1s"
fill="freeze"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <animate id="anim6b" targetElement="brush6" attributeName="height"
from="50" to="25" begin="anim6a.repeat(2)" dur="2s" fill="freeze"/>
      <animate id="anim7" targetElement="rect7" attributeName="height" from="50"
to="25" begin="accesskey(1)" dur="2s" fill="freeze"/>
      <animate id="anim8" targetElement="rect8" attributeName="height" from="50"
to="25" begin="wallclock(2000-01-01T00:00:00Z)" dur="2s" fill="freeze"/>
      <animate id="anim9" targetElement="rect9" attributeName="width" from="30"
to="30" begin="indefinite" dur="1s" fill="freeze"/>
    </par>
  </body>
</smil>
```

A Slide Part would reference this content as follows:

```
<p:spTree>
  …
  <p:contentPart r:id="smil01"/>
  …
</p:spTree>
```

The contentPart element specifies that the content targeted by the relationship with an ID of `smil01` is part of the PresentationML document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="smil01" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/customXml"
Target="smil1.xml" />
  …
</Relationships>
```

The corresponding relationship part item shows that the SMIL content is located next to the slide and is named `smil1.xml`. *end example*]

ttributes	Description
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a content part. [*Example*: Consider an XML element which has the following id attribute: `<… r:id="rId1" />` The markup specifies the associated relationship part with relationship ID `rId1` contains the corresponding relationship information for the parent XML element. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

ttributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.3. *end note*]

19.3.1.15 controls (List of controls)

This element specifies a list of embedded controls for the corresponding slide. Custom embedded controls can be embedded on slides.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ControlList) is located in §A.3. *end note*]

19.3.1.16 cSld (Common Slide Data)

This element specifies a container for slide information that is relevant to all of the slide types. All slides share a common set of properties that is independent of the slide type; the description of these properties for any particular slide is stored within the slide's cSld container. Slide data specific to the slide type indicated by the parent element is stored elsewhere.

[*Note*: The actual data in cSld describe only the particular parent slide; it is only the kind of information stored that is common across all slides. *end note*]

[*Example*: Consider the following PresentationML slide

```
<p:sld>
  <p:cSld>
    <p:spTree>
      …
    </p:spTree>
  </p:cSld>
  …
</p:sld>
```

As the above example shows, the shape tree of a slide (spTree) is a child element of cSld because all slide types can contain a shape tree. Other slide properties specific to the slide type (such as transitions for sld slides) are specified elsewhere. *end example*]

Attributes	Description
name (Name)	Specifies the slide name property that is used to further identify this unique configuration of common slide data. This might be used to aid in distinguishing different slide layouts or various other slide types. The possible values for this attribute are defined by the W3C XML Schema string

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommonSlideData) is located in §A.3. *end note*]

19.3.1.17 custData (Customer Data)

This element specifies customer data which allows for the specifying and persistence of customer specific data within the presentation.

Attributes	Description
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This attribute specifies the relationship id for referencing other resources outside the scope of the current PresentationML file. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomerData) is located in §A.3. *end note*]

19.3.1.18 custDataLst (Customer Data List)

This element allows for the specifying of customer defined data within the PresentationML framework. References to custom data or tags can be defined within this list.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomerDataList) is located in §A.3. *end note*]

19.3.1.19 cxnSp (Connection Shape)

This element specifies a connection shape that is used to connect two sp elements. Once a connection is specified using a cxnSp, it is left to the generating application to determine the exact path the connector takes. That is the connector routing algorithm is left up to the generating application as the desired path might be different depending on the specific needs of the application.

[*Example*: Consider the following connector shape that connects two regular shapes.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:spTree>
  …
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="1" name="Rectangle 1"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="2" name="Rectangle 2"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:cxnSp>
    <p:nvCxnSpPr>
      <p:cNvPr id="3" name="Elbow Connector 3"/>
      <p:cNvCxnSpPr>
        <a:stCxn id="1" idx="3"/>
        <a:endCxn id="2" idx="1"/>
      </p:cNvCxnSpPr>
      <p:nvPr/>
    </p:nvCxnSpPr>
    …
  </p:cxnSp>
</p:spTree>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connector) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.3.1.20 extLst (Extension List with Modification Flag)

This element specifies the extension list with modification ability within which all future extensions of element type ext are defined. The extension list along with corresponding future extensions is used to extend the storage capabilities of the PresentationML framework. This allows for various new kinds of data to be stored natively within the framework.

[*Note*: Using this extLst element allows the generating application to store whether this extension property has been modified. *end note*]

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
mod (Modify)	This attribute specifies whether the data contained within this element has been modified and should thus be processed again by the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExtensionListModify) is located in §A.3. *end note*]

19.3.1.21 graphicFrame (Graphic Frame)

This element specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.

Attributes	Description
bwMode (Black and White Mode) Namespace: .../drawingml/2006/main	Specifies how the graphical object should be rendered, using color, black or white, or grayscale. [*Note*: This does not mean that the graphical object itself is stored with only black and white or grayscale information. This attribute instead sets the rendering mode that the graphical object uses. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrame) is located in §A.3. *end note*]

19.3.1.22 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<p:grpSp>
  <p:nvGrpSpPr>
    <p:cNvPr id="10" name="Group 9"/>
    <p:cNvGrpSpPr/>
    <p:nvPr/>
  </p:nvGrpSpPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:grpSpPr>
  <a:xfrm>
    <a:off x="838200" y="990600"/>
    <a:ext cx="2426208" cy="978408"/>
    <a:chOff x="838200" y="990600"/>
    <a:chExt cx="2426208" cy="978408"/>
  </a:xfrm>
</p:grpSpPr>
<p:sp>
  …
</p:sp>
<p:sp>
  …
</p:sp>
<p:sp>
  …
</p:sp>
</p:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.3. *end note*]

19.3.1.23 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

19.3.1.24 handoutMaster (Handout Master)

This element specifies an instance of a handout master slide. Within a handout master slide are contained all elements that describe the objects and their corresponding formatting for within a handout slide. Within a handout master slide the cSld element specifies the common slide elements such as shapes and their attached text bodies. There are other properties within a handout master slide but cSld encompasses the majority of the intended purpose for a handoutMaster slide.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HandoutMaster) is located in §A.3. *end note*]

19.3.1.25 hf (Header/Footer information for a slide master)

This element specifies the header and footer information for a slide. Headers and footers consist of placeholders for text that should be consistent across all slides and slide types, such as a date and time, slide numbering, and custom header and footer text.

Attributes	Description
dt (Date/Time Placeholder)	Specifies whether the Date/Time placeholder is enabled for this master. If this attribute is not specified, a value of true should be assumed by the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
ftr (Footer Placeholder)	Specifies whether the Footer placeholder is enabled for this master. If this attribute is not specified, a value of true should be assumed by the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hdr (Header Placeholder)	Specifies whether the Header placeholder is enabled for this master. If this attribute is not specified, a value of true should be assumed by the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sldNum (Slide Number Placeholder)	Specifies whether the slide number placeholder is enabled. If this attribute is not specified, a value of true should be assumed by the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_HeaderFooter) is located in §A.3. *end note*]

19.3.1.26 notes (Notes Slide)

This element specifies the existence of a notes slide along with its corresponding data. Contained within a notes slide are all the common slide elements along with addition properties that are specific to the notes element.

[*Example*: Consider the following PresentationML notes slide

```
<p:notes>
  <p:cSld>
    …
  </p:cSld>
  …
</p:notes>
```

In the above example a notes element specifies the existence of a notes slide with all of its parts. Notice the cSld element, that specifies the common elements that can appear on any slide type and then any elements specify additional non-common properties for this notes slide. *end example*]

Attributes	Description
showMasterPhAnim (Show Master Placeholder Animations)	Specifies whether or not to display animations on placeholders from the master slide. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMasterSp (Show Master Shapes)	Specifies if shapes on the master slide should be shown on slides or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesSlide) is located in §A.3. *end note*]

19.3.1.27 notesMaster (Notes Master)

This element specifies an instance of a notes master slide. Within a notes master slide are contained all elements that describe the objects and their corresponding formatting for within a notes slide. Within a notes master slide the cSld element specifies the common slide elements such as shapes and their attached text bodies. There are other properties within a notes master slide but cSld encompasses the majority of the intended purpose for a notesMaster slide.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NotesMaster) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.3.1.28 notesStyle (Notes Text Style)

This element specifies the text formatting style for the all other text within a notes slide. This formatting is used on all text within the corresponding notes slides. The text formatting is specified by utilizing the DrawingML framework just as within a regular presentation slide. Within the notesStyle element there can be many different style types defined as there are different kinds of text stored within a notes slide.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

19.3.1.29 nvCxnSpPr (Non-Visual Properties for a Connection Shape)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorNonVisual) is located in §A.3. *end note*]

19.3.1.30 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrameNonVisual) is located in §A.3. *end note*]

19.3.1.31 nvGrpSpPr (Non-Visual Properties for a Group Shape)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeNonVisual) is located in §A.3. *end note*]

19.3.1.32 nvPicPr (Non-Visual Properties for a Picture)

This element specifies all non-visual properties for a picture. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following PresentationML.

```
<p:pic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    <p:nvPicPr>
      …
    </p:nvPicPr>
    …
  </p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.3. *end note*]

19.3.1.33 nvPr (Non-Visual Properties)

This element specifies non-visual properties for objects. These properties include multimedia content associated with an object and properties indicating how the object is to be used or displayed in different contexts.

Attributes	Description
isPhoto (Is a Photo Album)	Specifies whether the picture belongs to a photo album and should thus be included when editing a photo album within the generating application. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
userDrawn (Is User Drawn)	Specifies if the corresponding object has been drawn by the user and should thus not be deleted. This allows for the flagging of slides that contain user drawn data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ApplicationNonVisualDrawingProps) is located in §A.3. *end note*]

19.3.1.34 nvSpPr (Non-Visual Properties for a Shape)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeNonVisual) is located in §A.3. *end note*]

19.3.1.35 otherStyle (Slide Master Other Text Style)

This element specifies the text formatting style for the all other text within a master slide. This formatting is used on all text not covered by the titleStyle or bodyStyle elements within related presentation slides. The text formatting is specified by utilizing the DrawingML framework just as within a regular presentation slide. Within

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

the otherStyle element there can be many different style types defined as there are different kinds of text stored within a slide.

[*Note*: The otherStyle element is to be used for specifying the text formatting of text within a slide shape but not within a text box. Text box styling is handled from within the bodyStyle element. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

19.3.1.36 ph (Placeholder Shape)

This element specifies that the corresponding shape should be represented by the generating application as a placeholder. When a shape is considered a placeholder by the generating application it can have special properties to alert the user that they can enter content into the shape. Different placeholder types are allowed and can be specified by using the placeholder type attribute for this element.

Attributes	Description
hasCustomPrompt (Placeholder has custom prompt)	Specifies whether the corresponding placeholder should have a custom prompt or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
idx (Placeholder Index)	Specifies the placeholder index. This is used when applying templates or changing layouts to match a placeholder on one template/master to another. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
orient (Placeholder Orientation)	Specifies the orientation of a placeholder. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).
sz (Placeholder Size)	Specifies the size of a placeholder. The possible values for this attribute are defined by the ST_PlaceholderSize simple type (§19.7.9).
type (Placeholder Type)	Specifies what content type a placeholder is intended to contain. The possible values for this attribute are defined by the ST_PlaceholderType simple type (§19.7.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Placeholder) is located in §A.3. *end note*]

19.3.1.37 pic (Picture)

This element specifies the existence of a picture object within the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following PresentationML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  <p:blipFill>
    …
  </p:blipFill>
  <p:spPr>
    …
  </p:spPr>
</p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.3. *end note*]

19.3.1.38 sld (Presentation Slide)

This element is the root element of a Slide part (§13.3.8) and specifies an instance of a slide. Within a slide are contained all elements that describe the objects and their corresponding formatting within a presentation slide. Child elements describe the common slide elements such as shapes and their attached text bodies, transition and timing specific to this slide and color information specific to this slide.

[*Example*: Consider the following basic slide.

```
<p:sld>
      <p:cSld>
            <p:spTree>
                  …
            </p:spTree>
      </p:cSld>
      <p:clrMapOver>
            …
      </p:clrMapOver>
      <p:transition>
            …
      </p:transition>
      <p:timing>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            …
        </p:timing>
    </p:sld>
```

This example shows a slide with its content in the shape tree, a local color mapping override and a slide transition with associated timing information. *end example*]

19.3.1.39 sldLayout (Slide Layout)

This element specifies an instance of a slide layout. The slide layout contains in essence a template slide design that can be applied to any existing slide. When applied to an existing slide all corresponding content should be mapped to the new slide layout.

Attributes	Description
matchingName (Matching Name)	Specifies a name to be used in place of the name attribute within the cSld element. This is used for layout matching in response to layout changes and template applications. The possible values for this attribute are defined by the W3C XML Schema string datatype.
preserve (Preserve Slide Layout)	Specifies whether the corresponding slide layout is deleted when all the slides that follow that layout are deleted. If this attribute is not specified then a value of false should be assumed by the generating application. This would mean that the slide would in fact be deleted if no slides within the presentation were related to it. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMasterPhAnim (Show Master Placeholder Animations)	Specifies whether or not to display animations on placeholders from the master slide. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
showMasterSp (Show Master Shapes)	Specifies if shapes on the master slide should be shown on slides or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
type (Slide Layout Type)	Specifies the slide layout type that is used by this slide. The possible values for this attribute are defined by the ST_SlideLayoutType simple type (§19.7.15).
userDrawn (Is User Drawn)	Specifies if the corresponding object has been drawn by the user and should thus not be deleted. This allows for the flagging of slides that contain user drawn data. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideLayout) is located in §A.3. *end note*]

19.3.1.40 sldLayoutId (Slide Layout Id)

This element specifies the relationship information for each slide layout that is used within the slide master. The slide master has relationship identifiers that it uses internally for determining the slide layouts that should be used. Then, to resolve what these slide layouts should be the sldLayoutId elements in the sldLayoutIdLst are utilized.

Attributes	Description
id (ID Tag) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id value that the generating application can use to resolve which slide layout is used in the creation of the slide. This relationship id is used within the relationship file for the master slide to expose the location of the corresponding layout file within the presentation. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
id (ID Tag)	Specifies the identification number that uniquely identifies this slide layout within the presentation file. The possible values for this attribute are defined by the ST_SlideLayoutId simple type (§19.7.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideLayoutIdListEntry) is located in §A.3. *end note*]

19.3.1.41 sldLayoutIdLst (List of Slide Layouts)

This element specifies the existence of the slide layout identification list. This list is contained within the slide master and is used to determine which layouts are being used within the slide master file. Each layout within the list of slide layouts has its own identification number and relationship identifier that uniquely identifies it within both the presentation document and the particular master slide within which it is used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideLayoutIdList) is located in §A.3. *end note*]

19.3.1.42 sldMaster (Slide Master)

This element specifies an instance of a slide master slide. Within a slide master slide are contained all elements that describe the objects and their corresponding formatting for within a presentation slide. Within a slide master slide are two main elements. The cSld element specifies the common slide elements such as shapes and their attached text bodies. Then the txStyles element specifies the formatting for the text within each of these shapes. The other properties within a slide master slide specify other properties for within a presentation slide such as color information, headers and footers, as well as timing and transition information for all corresponding presentation slides.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
preserve (Preserve Slide Master)	Specifies whether the corresponding slide layout is deleted when all the slides that follow that layout are deleted. If this attribute is not specified then a value of false should be assumed by the generating application. This would mean that the slide would in fact be deleted if no slides within the presentation were related to it. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideMaster) is located in §A.3. *end note*]

19.3.1.43 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the DrawingML framework. In addition to a geometry each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

[*Note*: Shapes are the preferred mechanism for specifying text on a slide. *end note*]

Attributes	Description
useBgFill (Use Background Fill)	Specifies that the shape fill should be set to that of the slide background surface. [*Note*: This attribute does not set the fill of the shape to be transparent but instead sets it to be filled with the portion of the slide background that is directly behind it. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.3. *end note*]

19.3.1.44 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a shape. These properties include the shape fill, outline, geometry, effects, and 3D orientation.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

19.3.1.45 spTree (Shape Tree)

This element specifies all shape-based objects, either grouped or not, that can be referenced on a given slide. As most objects within a slide are shapes, this represents the majority of content within a slide. Text and effects are attached to shapes that are contained within the spTree element.

[*Example*: Consider the following PresentationML slide

```
<p:sld>
  <p:cSld>
    <p:spTree>
      <p:nvGrpSpPr>
        …
      </p:nvGrpSpPr>
      <p:grpSpPr>
        …
      </p:grpSpPr>
      <p:sp>
        …
      </p:sp>
    </p:spTree>
  </p:cSld>
  …
</p:sld>
```

In the above example the shape tree specifies all the shape properties for this slide. *end example*]

Each shape-based object within the shape tree, whether grouped or not, shall represent one unique level of z-ordering on the slide. The z-order for each shape-based object shall be determined by the lexical ordering of each shape-based object within the shape tree: the first shape-based object shall have the lowest z-order, while the last shape-based object shall have the highest z-order.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The z-ordering of shape-based objects within the shape tree shall also determine the navigation (tab) order of the shape-based objects: the shape-based object with the lowest z-order (the first shape in lexical order) shall be first in navigation order, with objects being navigated in ascending z-order.

[*Example*: Consider the following PresentationML slide with two shapes

```
<p:sld>
  <p:cSld>
    <p:spTree>
      …
      <p:sp>
        <p:nvSpPr>
          <p:cNvPr id="5" name="Oval 4" />
          …
        </p:nvSpPr>
        …
      </p:sp>
      <p:sp>
        <p:nvSpPr>
          <p:cNvPr id="4" name="Isosceles Triangle 3" />
          …
        </p:nvSpPr>
        …
      </p:sp>
    </p:spTree>
  </p:cSld>
  …
</p:sld>
```

In the above example the shape with name `Oval 4` has the lowest z-order value since that shape is the first shape in the shape tree. `Oval 4` is also the first shape in navigation order. The shape with name `Isosceles Triangle 3` has the highest z positioning value since that shape is the last shape in the shape tree. `Isosceles Triangle 3` is also the last shape in navigation order. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.3.1.46 style (Shape Style)

This element specifies the style information for a shape. This is used to define a shape's appearance in terms of the preset styles defined by the style matrix for the theme.

[*Example*:

```
<p:style>
  <a:lnRef idx="3">
```

```
      <a:schemeClr val="lt1"/>
    </a:lnRef>
    <a:fillRef idx="1">
      <a:schemeClr val="accent3"/>
    </a:fillRef>
    <a:effectRef idx="1">
      <a:schemeClr val="accent3"/>
    </a:effectRef>
    <a:fontRef idx="minor">
      <a:schemeClr val="lt1"/>
    </a:fontRef>
  </p:style>
```

The parent shape of the above code is to have an outline that uses the third line style defined by the theme, use the first fill defined by the scheme, and be rendered with the first effect defined by the theme. Text inside the shape is to use the minor font defined by the theme.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

19.3.1.47 tags (Customer Data Tags)

This element specifies the existence of customer data in the form of tags. This allows for the storage of customer data within the PresentationML framework. While this is similar to the ext tag in that it can be used store information, this tag mainly focuses on referencing to other parts of the presentation document. This is accomplished via the relationship identification attribute that is required for all specified tags.

Attributes	Description
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	This attribute specifies the relationship identifier for the customer data tag. This allows for a link to a resource that is external from the current XML document but still contained within the presentation document. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TagsData) is located in §A.3. *end note*]

19.3.1.48 timing (Slide Timing Information for a Slide Layout)

This element specifies the timing information for handling all animations and timed events within the corresponding slide. This information is tracked via time nodes within the timing element. More information on

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

the specifics of these time nodes and how they are to be defined can be found within the Animation section of the PresentationML framework.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideTiming) is located in §A.3. *end note*]

19.3.1.49 titleStyle (Slide Master Title Text Style)

This element specifies the text formatting style for the title text within a master slide. This formatting is used on all title text within related presentation slides. The text formatting is specified by utilizing the DrawingML framework just as within a regular presentation slide. Within a title style there can be many different style types defined as there are different kinds of text stored within a slide title.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

19.3.1.50 transition (Slide Transition for a Slide Layout)

This element specifies the kind of slide transition that should be used to transition to the current slide from the previous slide. That is, the transition information is stored on the slide that appears after the transition is complete.

Attributes	Description
advClick (Advance on Click)	Specifies whether a mouse click advances the slide or not. If this attribute is not specified then a value of true is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
advTm (Advance after time)	Specifies the time, in milliseconds, after which the transition should start. This setting can be used in conjunction with the advClick attribute. If this attribute is not specified then it is assumed that no auto-advance occurs. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spd (Transition Speed)	Specifies the transition speed that is to be used when transitioning from the current slide to the next. The possible values for this attribute are defined by the ST_TransitionSpeed simple type (§19.7.54).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideTransition) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.3.1.51 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

19.3.1.52 txStyles (Slide Master Text Styles)

This element specifies the text styles within a slide master. Within this element is the styling information for title text, the body text and other slide text as well. This element is only for use within the Slide Master and thus sets the text styles for the corresponding presentation slides.

[*Example*: Consider the case where we would like to specify the title text for a master slide.

```
<p:txStyles>
  <p:titleStyle>
    <a:lvl1pPr algn="ctr" rtl="0" latinLnBrk="0">
      <a:spcBef>
        <a:spcPct val="0"/>
      </a:spcBef>
      <a:buNone/>
      <a:defRPr sz="4400" kern="1200">
        <a:solidFill>
          <a:schemeClr val="tx1"/>
        </a:solidFill>
        <a:latin typeface="+mj-lt"/>
        <a:ea typeface="+mj-ea"/>
        <a:cs typeface="+mj-cs"/>
      </a:defRPr>
    </a:lvl1pPr>
  </p:titleStyle>
</p:txStyles>
```

In the above example the title text is set according to the above formatting for all related slides within the presentation. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideMasterTextStyles) is located in §A.3. *end note*]

19.3.1.53 xfrm (2D Transform for Graphic Frame)

This element specifies the transform to be applied to the corresponding graphic frame. This transformation is applied to the graphic frame just as it would be for a shape or group shape.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

19.3.2 Embedded Objects

Within the slides portion of PresentationML, there are the embedded elements. These are objects that can be embedded within a slide. As we defined a slide to be a container it can be seen that it does not just contain shapes, pictures and text but embedded objects as well that are not necessarily native to the PresentationML platform.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.3.2.1 control (Embedded Control)

This element specifies the existence of an embedded control in the slide.

Attributes	Description
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id that is used to identify this Embedded object from within a slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
imgH (Image Height)	Specifies the height of the embedded control. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).
imgW (Image Width)	Specifies the width of the embedded control. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).
name (Embedded Object Name)	Specifies the identifying name class used by scripting languages. This name is also used to construct the clipboard name. The possible values for this attribute are defined by the W3C XML Schema string datatype.
showAsIcon (Show Embedded Object As Icon)	Specifies whether the Embedded object shows as an icon or using its native representation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Control) is located in §A.3. *end note*]

19.3.2.2 embed (Embedded Object or Control)

This element specifies an Embedded object or Control that is embedded within the presentation.

Attributes	Description
followColorScheme (Color Scheme Properties for Embedded object)	Specifies the Color Scheme Properties for the corresponding Embedded object being specified. The possible values for this attribute are defined by the ST_OleObjectFollowColorScheme simple type (§19.7.6).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleObjectEmbed) is located in §A.3. *end note*]

19.3.2.3 link (Linked Object or Control)

This element specifies a link to an external Embedded object or Control.

Attributes	Description
updateAutomatic (Update Linked Embedded Objects Automatically)	This attribute determines if linked embedded objects are automatically updated when the presentation is opened or printed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleObjectLink) is located in §A.3. *end note*]

19.3.2.4 oleObj (Global Element for Embedded objects and Controls)

This element specifies a global element to be used for an Embedded object and Control.

When the oleObject element contains a pic child element, the identifier specified by the pic/nvPicPr/cNvPr@id attribute shall be ignored and the identifier specified by the graphicFrame/nvGraphicFramePr/cNvPr@id attribute shall be used when deciding which identifier to use for the OLE object.

Attributes	Description
id (Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id that is used to identify this Embedded object from within a slide. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
imgH (Image Height)	Specifies the height of the embedded control. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).
imgW (Image Width)	Specifies the width of the embedded control. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).
name (Embedded Object Name)	Specifies the identifying name class used by scripting languages. This name is also used to construct the clipboard name.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
progId (Embedded Object ProgID)	Specifies the progid for an Embedded object. The possible values for this attribute are defined by the W3C XML Schema string datatype.
showAsIcon (Show Embedded Object As Icon)	Specifies whether the Embedded object shows as an icon or using its native representation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OleObject) is located in §A.3. *end note*]

19.3.3 Programmable Tags

Within the slides portion of PresentationML there are the tag elements. These are extensibility names and values that assist in the storage of legacy variables from older file formats.

19.3.3.1 tag (Programmable Extensibility Tag)

This element specifies a programmable extensibility tag to be used for storage of legacy variables.

Attributes	Description
name (Name)	Specifies the name associated with this specific programmable tag. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Value)	Specifies the value associated with this specific programmable tag. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StringTag) is located in §A.3. *end note*]

19.3.3.2 tagLst (Programmable Tab List)

This element specifies the list of programmable extensibility tags that are used to store variables from legacy file formats.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TagList) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.4 Comments

A comment is a text note attached to a slide, with the primary purpose of allowing readers of a presentation to provide feedback to the presentation author. Each comment contains an unformatted text string and information about its author, and is attached to a particular location on a slide. Comments can be visible while editing the presentation, but do not appear when a slide show is given. The displaying application decides when to display comments and determines their visual appearance.

19.4.1 cm (Comment)

This element specifies a single comment attached to a slide. It contains the text of the comment, its position on the slide, and attributes referring to its author and date.

[*Example:*

```
<p:cm authorId="0" dt="2006-08-28T17:26:44.129" idx="1">
  <p:pos x="10" y="10"/>
  <p:text>Add diagram to clarify.</p:text>
</p:cm>
```

end example]

Attributes	Description
authorId (Comment Author ID)	This attribute specifies the author of the comment. It refers to the ID of an author in the comment author list for the document. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
dt (Comment Date/Time)	This attribute specifies the date and time this comment was last modified. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
idx (Comment Index)	This attribute specifies an identifier for this comment that is unique within a list of all comments by this author in this document. An author's first comment in a document has index 1. [*Note:* Because the index is unique only for the comment author, a document can contain multiple comments with the same index created by different authors. *end note*] The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Comment) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.4.2 cmAuthor (Comment Author)

This element specifies a single author with comments in the document. It contains a unique author ID, the author's name and initials, the index of the author's last comment, and the index of a color associated with the author.

[*Example:*

```
<p:cmAuthor id="0" name="Julie Lee" initials="JL" lastIdx="1" clrIdx="0"/>
```

end example]

Attributes	Description
clrIdx (Comment Author Color Index)	This attribute specifies an index into the generating application's comments color table to allow for visual (color) differentiation of different author's comments. This color is used for all comments by this author. If more authors exist than there are entries in the color table, the color index wraps around to the beginning of the table. [*Note:* It is left entirely up to the generating application to determine the amount of colors used in the comments color table and in what order these are used when rendering comments on a slide surface. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (Comment Author ID)	This attribute specifies a unique (within the document) zero-based identifier that refers to a single comment author. [*Note:* The method of generating an author id is determined by the application and need not be sequential, provided each id is unique within the list of comment authors for the document. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
initials (Comment Author Initials)	This attribute specifies a string that represents the initials of this particular author. The value is not necessarily unique. It is intended for use by the application as an abbreviated version of the comment author's name. The possible values for this attribute are defined by the ST_Name simple type (§19.7.5).
lastIdx (Index of Comment Author's last comment)	Index of the last comment added to this document by this author. New comments by this author are counted starting with the value one greater than this index. [*Note:* The index of a deleted comment is not reused; therefore, this value is not an accurate count of the total number of comments by the author. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
name (Comment Author Name)	This attribute specifies the full name of this particular author. As a string, it has no security or authentication data. This value is not guaranteed to be unique across all document authors. The possible values for this attribute are defined by the ST_Name simple type (§19.7.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommentAuthor) is located in §A.3. *end note*]

19.4.3 cmAuthorLst (List of Comment Authors)

This element specifies a list of authors with comments in the current document. Each comment in a document shall refer to an author in this list. No cmAuthor element in a cmAuthorLst shall have both the same name attribute value and the same initials attribute value as any other cmAuthor element in the same cmAuthorLst.

[*Example:* A document contains comments left by two authors.

```
<p:cmAuthorLst>
    <p:cmAuthor id="0" name="Julie Lee" initials="JL" lastIdx="1" clrIdx="0"/>
    <p:cmAuthor id="1" name="Fred Jones" initials="FJ" lastIdx="2" clrIdx="1"/>
</p:cmAuthorLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommentAuthorList) is located in §A.3. *end note*]

19.4.4 cmLst (Comment List)

This element specifies a list of comments for a particular slide.

[*Example:* A slide contains two comments, each left by a different author. This example demonstrates that two comments can have the same index if they are created by different authors.

```
<p:cmLst>
    <p:cm authorId="0" dt="2006-08-28T17:26:44.129" idx="1">
        <p:pos x="10" y="10"/>
        <p:text>Add diagram to clarify.</p:text>
    </p:cm>
    <p:cm authorId="1" dt="2006-08-28T17:44:19.679" idx="1">
        <p:pos x="1426" y="660"/>
        <p:text>Clean up this text.</p:text>
    </p:cm>
</p:cmLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CommentList) is located in §A.3. *end note*]

19.4.5 pos (Comment Position)

This element specifies the positioning information for the placement of a comment on a slide surface. In LTR versions of the generating application, this position information should refer to the upper left point of the comment shape. In RTL versions of the generating application, this position information should refer to the upper right point of the comment shape.

[*Note:* The anchoring point on the slide surface is unaffected by a right-to-left or left-to-right layout change. That is the anchoring point remains the same for all language versions. *end note*]

[*Note:* Because there is no specified size or formatting for comments, this UI widget used to display a comment can be any size and thus the lower right point of the comment shape is determined by how the viewing application chooses to display comments. *end note*]

[*Example:*

```
<p:pos x="1426" y="660"/>
```

end example]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the y-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

19.4.6 text (Comment's Text Content)

This element specifies the content of a comment. This is the text with which the author has annotated the slide.

[*Example:*

```
<p:text>Add diagram to clarify.</p:text>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.5 Animation

The Animation section of the PresentationML framework stores the movement and related information of objects.

This schema is loosely based on the syntax and concepts from the Synchronized Multimedia Integration Language (SMIL), a W3C Recommendation for describing multimedia presentations using XML.

The schema describes all the animations effects that reside on a slide and also the animation that occurs when going from slide to slide (slide transition).

Animations on a slide are inherently time-based and consist of an animation effects on an object or text. Slide transitions however do not follow this concept and always appear before any animation on a slide.

All elements described in this schema are contained within the slide XML file. More specifically they are in the <transition> and the <timing> element as shown below:

```
<p:sld>
  <p:cSld> … </p:cSld>
  <p:clrMapOvr> … </p:clrMapOvr>
  <p:transition> … </p:transition>
  <p:timing> … </p:timing>
</p:sld>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.1 anim (Animate)

This element is a generic animation element that requires little or no semantic understanding of the attribute being animated. It can animate text within a shape or even the shape itself.

[*Example*: Consider trying to emphasize text within a shape by changing the size of its font by 150%. The <anim> element should be used as follows:

```
<p:anim to="1.5" calcmode="lin" valueType="num">
  <p:cBhvr override="childStyle">
    <p:cTn id="1" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="1">
        <p:txEl>
          <p:charRg st="1" end="4"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>style.fontSize</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:anim>
```

end example]

Attributes	Description
by (By)	This attribute specifies a relative offset value for the animation with respect to its position before the start of the animation. The possible values for this attribute are defined by the W3C XML Schema string datatype.
calcmode (Calculation Mode)	This attribute specifies the interpolation mode for the animation. The possible values for this attribute are defined by the ST_TLAnimateBehaviorCalcMode simple type (§19.7.20).
from (From)	This attribute specifies the starting value of the animation. The possible values for this attribute are defined by the W3C XML Schema string datatype.
to (To)	This attribute specifies the ending value for the animation as a percentage. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
valueType (Value Type)	This attribute specifies the type of property value. The possible values for this attribute are defined by the ST_TLAnimateBehaviorValueType simple type (§19.7.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimateBehavior) is located in §A.3. *end note*]

19.5.2 animClr (Animate Color Behavior)

This animation element is responsible for animating the color of an object.

[*Example*: Consider trying to emphasize a shape by changing its fill color to scheme color accent2. The <animClr> element should be used as follows:

```
<p:animClr clrSpc="rgb">
  <p:cBhvr>
    <p:cTn id="1" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="1"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>fillcolor</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:to>
    <a:schemeClr val="accent2"/>
  </p:to>
</p:animClr>
```

end example]

Attributes	Description
clrSpc (Color Space)	This attribute specifies the color space in which to interpolate the animation. Values for example can be HSL & RGB. The values for from/to/by/etc. can still be specified in any supported color format without affecting the color space within which the animation happens. The RGB color space is best used for doing animations between two different colors since it doesn't require going through any other hues between the two colors specified. The HSL space is useful for animating through a rainbow of colors or for modifying just the saturation by 30% for example.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TLAnimateColorSpace simple type (§19.7.23).
dir (Direction)	This attribute specifies which direction to cycle the hue around the color wheel. Values are clockwise or counter clockwise. Default is clockwise. The possible values for this attribute are defined by the ST_TLAnimateColorDirection simple type (§19.7.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimateColorBehavior) is located in §A.3. *end note*]

19.5.3 animEffect (Animate Effect)

This animation behavior provides the ability to do image transform/filter effects on elements. Some visual effects are dynamic in nature and have a progress that animates from 0 to 1 over a period of time to do visual transitions between hidden and visible states. Other filters are static and apply a effects like a blur or drop-shadow which aren't inherently time-based.

[*Example*: Consider trying to emphasize a shape by creating an entrance animation using a "blinds" motion.

```
<p:animEffect transition="in" filter="blinds(horizontal)">
  <p:cBhvr>
    <p:cTn id="7" dur="500"/>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

end example]

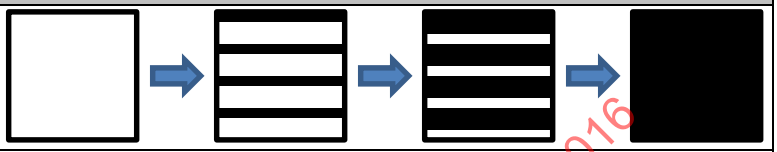
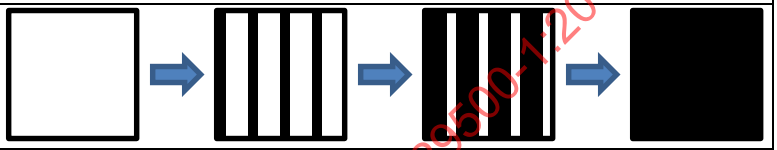
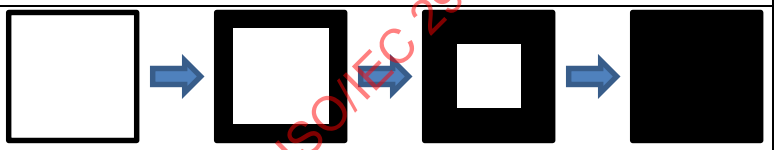
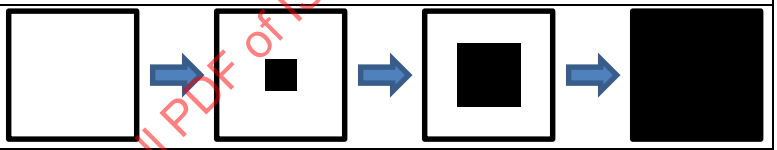
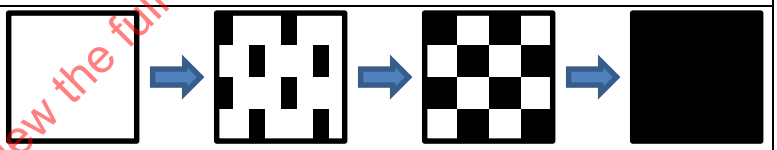
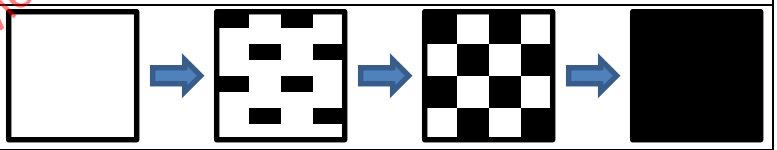
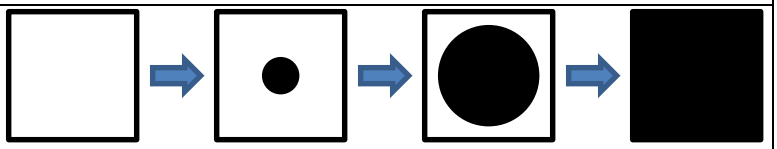
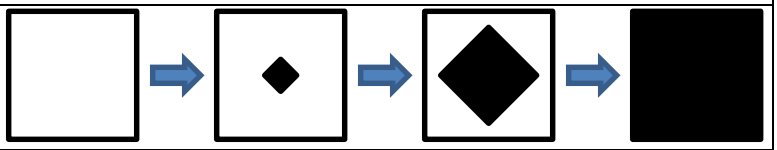
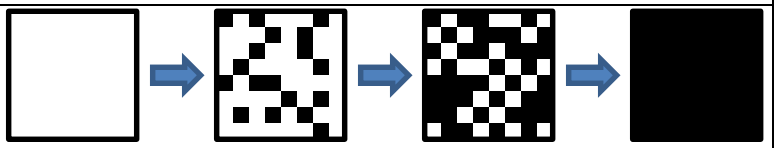
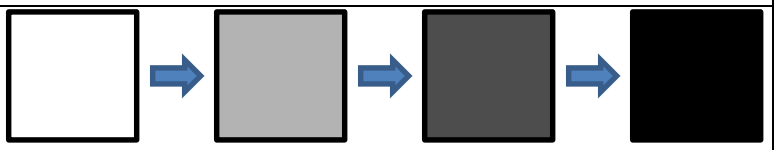
Attributes	Description
filter (Filter)	This attribute specifies the animation types and subtypes to be used for the effect. Multiple animations are allowed to be listed so that in the event that a superseding animation (leftmost) cannot be rendered, a fallback animation is available. That is, the rendering application parses the list from left to right until a supported animation is found. The syntax used for the filter attribute value is as follows: "type(subtype);type(subtype)". Subtype can be a string value such as "fromLeft" or a numerical value depending on the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

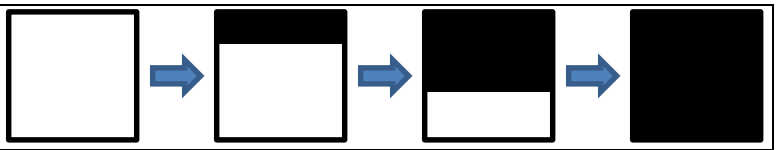
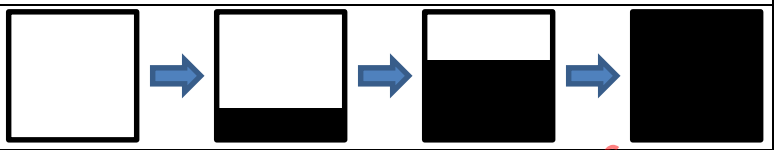
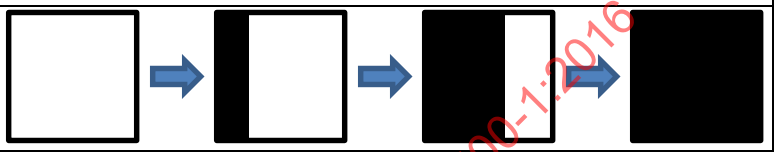
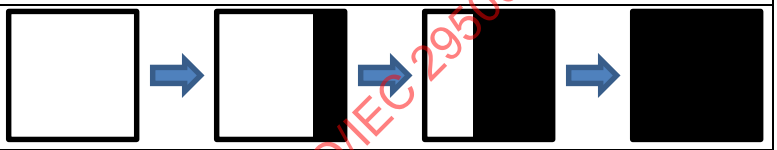
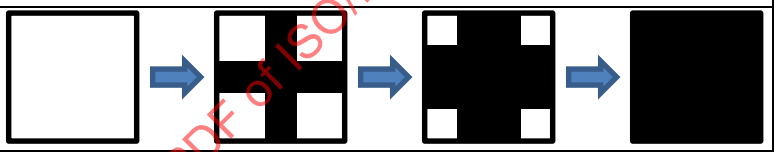
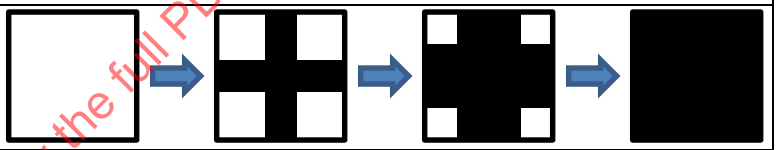
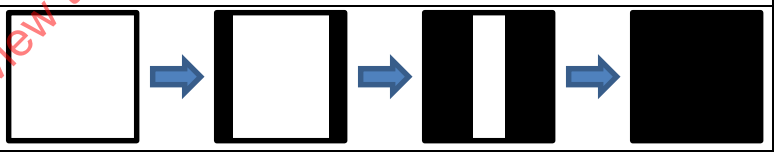
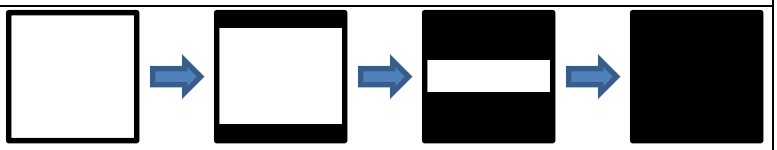
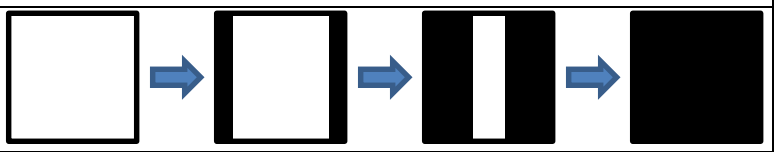
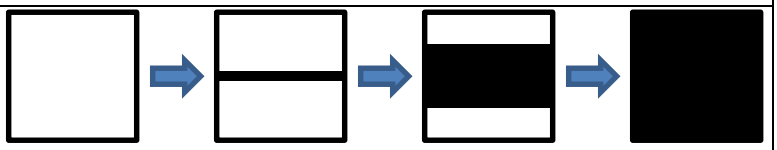
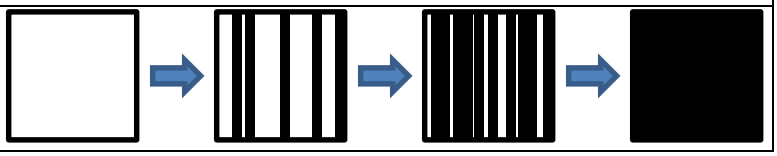
<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>type specified.

Reserved Animation Types(subtypes):

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>blinds(horizontal)</td><td>  </td></tr>
<tr><td>blinds(vertical)</td><td>  </td></tr>
<tr><td>box(in)</td><td>  </td></tr>
<tr><td>box(out)</td><td>  </td></tr>
<tr><td>checkerboard(across)</td><td>  </td></tr>
<tr><td>checkerboard(down)</td><td>  </td></tr>
<tr><td>circle</td><td>  </td></tr>
<tr><td>diamond</td><td>  </td></tr>
<tr><td>dissolve</td><td>  </td></tr>
<tr><td>fade</td><td>  </td></tr>
</table>
</td></tr>
</table>

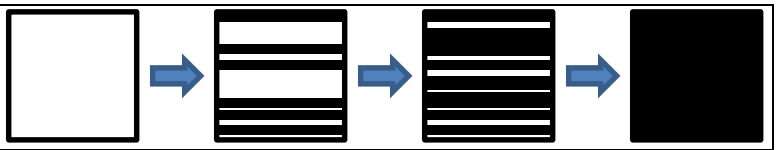
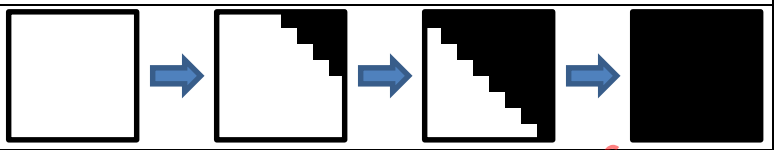
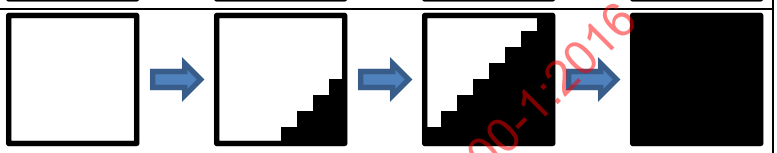
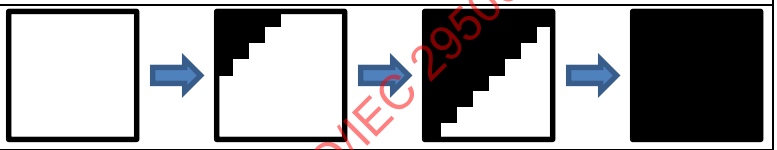
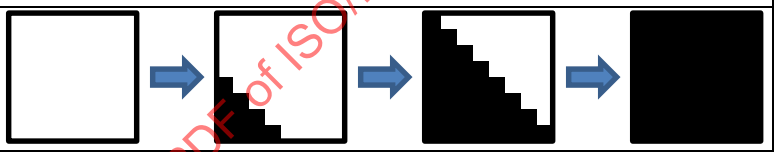
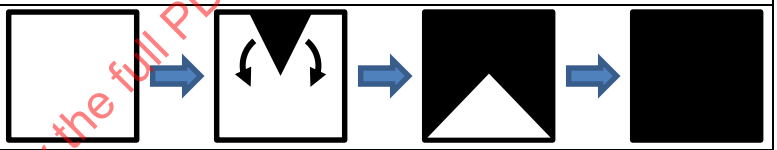
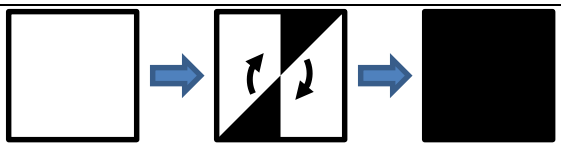
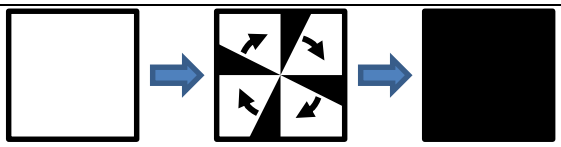
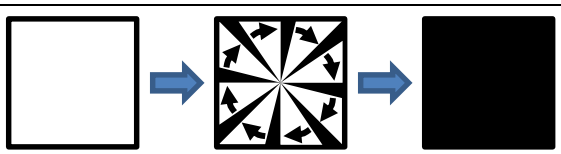
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description	
	slide(fromTop)	
	slide(fromBottom)	
	slide(fromLeft)	
	slide(fromRight)	
	plus(in)	
	plus(out)	
	barn(inVertical)	
	barn(inHorizontal)	
	barn(outVertical)	
	barn(outHorizontal)	
	randomBars(horizontal)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

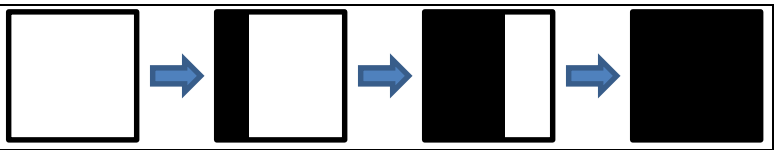
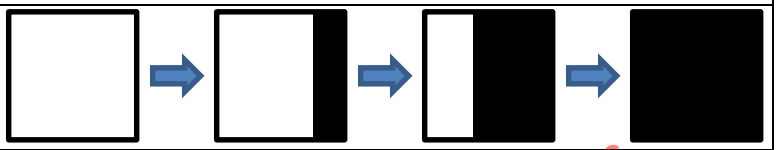
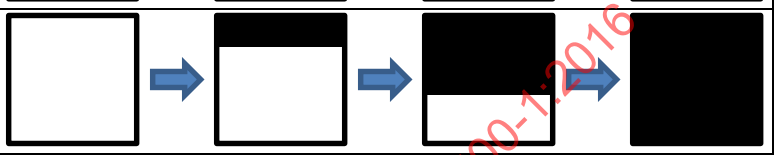
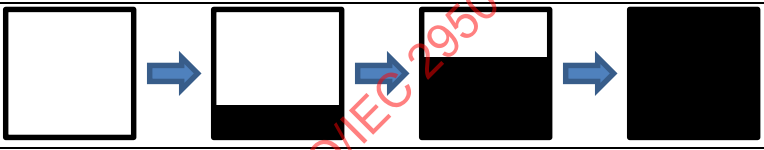
©ISO/IEC 2016 – All rights reserved

Attributes	Description
	randomBars(vertical) 
	strips(downLeft) 
	strips(upLeft) 
	strips(downRight) 
	strips(upRight) 
	wedge 
	wheel(1)
	wheel(2) 
	wheel(3)
	wheel(4) 
	wheel(8) 

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	wipe(right) 
	wipe(left) 
	wipe(down) 
	wipe(up) 
	
	[*Note:* The renderings shown above are for example purposes only. Exact rendering of any animation is determined by the rendering application. As such, the same animation can have many variations, depending on the implementation. More detail for each rendering above can be found in transition (§19.3.1.50). *end note*]
	
	[*Example*: Consider the following animation effect:
	
	`<p:animEffect transition="in"`
	`filter="blinds(horizontal);blinds(vertical)">`
	`<p:cBhvr>`
	`<p:cTn id="7" dur="500"/>`
	`<p:tgtEl>`
	`<p:spTgtspid="5"/>`
	`</p:tgtEl>`
	`</p:cBhvr>`
	`</p:animEffect>`
	
	There are two animation filters shown in this example. The first is the blinds (horizontal), which the rendering application is to use as the primary animation effect. If, however, the rendering application does not support this animation, the blinds (vertical) animation is used. In this example there are only two animation filters listed, a primary and a fallback, but it is possible to list multiple fallback filters using the syntax defined above.
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

prLst (Property List)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	This attribute specifies a list of properties that coincide with the effect specified. Although there are many animation types allowed, this attribute allows the setting of specific property settings in order to describe an even wider variety of animation types.
	
	The syntax used for the prLst attribute value is as follows: “name:value;name:value”. When multiple animation types are listed in the filter attribute, the rendering application attempts to apply each property value even though some might not apply to it.
	
	Reserved Names(values):
	• opacity (float values of 0.0 - 1.0)
	
	[*Example*: Consider the following animation effect:
	
	`<p:animEffect filter="image" prLst="opacity: 0.5">`
	`<p:cBhvr rctx="IE">`
	`<p:cTn id="7" dur="indefinite"/>`
	`<p:tgtEl>`
	`<p:spTgtspid="3"/>`
	`</p:tgtEl>`
	`</p:cBhvr>`
	`</p:animEffect>`
	
	The animation filter specified is an image filter type that has a specific property called opacity set to a value of 0.5. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

transition (Transition)	This attribute specifies whether to transition the element in or out or treat it as a static filter. The values are "none", "in" and "out", and the default value is "in". When a value of "in" is specified, the element is not visible at the start of the animation and is completely visible be the end of the duration. When "out" is specified, the element is visible at the start and not visible at the end of the effect. This visibility is in addition to the effect of setting CSS visibility or display attributes. The possible values for this attribute are defined by the ST_TLAnimateEffectTransition simple type (§19.7.24).

[*Note*: The W3C XML Schema definition of this element’s content model (CT_TLAnimateEffectBehavior) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.4 animMotion (Animate Motion)

Animate motion provides an abstracted way to move positioned elements. It provides the ability to specify from/to/by motion as well as to use more detailed path descriptions for motion over polylines or bezier curves.

[*Example*: Consider animating a shape from its original position to the right.. The <animMotion> element should be used as follows:

```
<p:animMotion origin="layout" path="M 0 0 L 0.25 0 E" pathEditMode="relative">
  <p:cBhvr>
    <p:cTn id="1" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="1"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>ppt_x</p:attrName>
      <p:attrName>ppt_y</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:animMotion>
```

end example]

Attributes	Description
origin (Origin)	Specifies what the origin of the motion path is relative to such as the layout of the slide, or the parent. The possible values for this attribute are defined by the ST_TLAnimateMotionBehaviorOrigin simple type (§19.7.25).
path (Path)	Specifies the path primitive followed by coordinates for the animation motion. The allowed values that are understood within a path are as follows: M = move to, L = line to, C = curve to, Z=close loop, E=end UPPERCASE = absolute coords, lowercase = relative coords Thus total allowed set = {M,L,C,Z,E,m,l,c,z,e) [*Example*: The following string is a sample path. path: “M 0 0 L 1 1 c 1 2 3 4 4 4 Z” *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
pathEditMode (Path Edit Mode)	This attribute specifies how the motion path moves when the target element is moved. The possible values for this attribute are defined by the ST_TLAnimateMotionPathEditMode simple type (§19.7.26).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
ptsTypes (Points Types)	This attribute describes the point type of the points in the path attribute. The allowed values that are understood for the ptsTypes attribute are as follows: A = Auto, F = Corner, T = Straight, S = Smooth UPPERCASE = Straight Line follows point, lowercase = curve follows point. Thus, the total allowed set = {A,F,T,S,a,f,t,s} The possible values for this attribute are defined by the W3C XML Schema string datatype.
rAng (Relative Angle)	The attribute describes the relative angle of the motion path. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimateMotionBehavior) is located in §A.3. *end note*]

19.5.5 animRot (Animate Rotation)

This animation element is responsible for animating the rotation of an object. Rotation values set in the "by" , "to, and "from" attributes are specified in degrees measured to a 60,000th, i.e 1 degree is 60,000. Rotation values can be larger than 360°.

The sign of the rotation angle specifies the direction for rotation. A negative rotation specifies that the rotation should appear in the host to go counter-clockwise".

[*Example*: Consider trying to emphasize a shape by rotating it 360 degrees clockwise. The <animRot> element should be used as follows:

```
<p:animRot by="21600000">
  <p:cBhvr>
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="5"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>r</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:animRot>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
by (By)	This attribute describes the relative offset value for the animation. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).
from (From)	This attribute describes the starting value for the animation. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).
to (To)	This attribute describes the ending value for the animation. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimateRotationBehavior) is located in §A.3. *end note*]

19.5.6 animScale (Animate Scale)

This animation element is responsible for animating the scale of an object. When animating the scale, the element shall scale around the reference point of the element and the positioning system used should be consistent with the one used for motion paths. When animating the width and height of an element, all of the width/height animation values are calculated first then the scale animations are applied on top of that. So for example, an animation from 0 to 100 of the width with a concurrent scale from 100% to 200% would result in the element appearing to scale from 0 to 200.

[*Example*: Consider trying to emphasize a shape by scaling it larger by 150%. The <animScale> element should be used as follows:

```
<p:childTnLst>
  <p:animScale>
    <p:cBhvr>
      <p:cTn id="6" dur="2000" fill="hold"/>
      <p:tgtEl>
        <p:spTgt spid="5"/>
      </p:tgtEl>
    </p:cBhvr>
    <p:by x="150000" y="150000"/>
  </p:animScale>
</p:childTnLst>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
zoomContents (Zoom Content)	This attribute specifies whether to zoom the contents of an object when doing a scaling animation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimateScaleBehavior) is located in §A.3. *end note*]

19.5.7 attrName (Attribute Name)

This element is used to contain an attribute value for an Attribute Name List. This value defines the specific attribute that an animation should be applied to, such as fill, style, and shadow, etc. A specific property is defined by using a "property.sub-property" format which is often extended to multiple sub properties as seen in the allowed values below.

Allowed property values:

```
style.opacity, style.rotation, style.visibility, style.color, style.fontSize,
style.fontWeight, style.fontStyle, style.fontFamily, style.textEffectEmboss,
style.textShadow, style.textTransform, style.textDecorationUnderline,
style.textEffectOutline, style.textDecorationLineThrough, style.sRotation,
imageData.cropTop, imageData.cropBottom, imageData.cropLeft,
imageData.cropRight, imageData.cropRight, imageData.gain, imageData.blacklevel,
imageData.gamma, imageData.grayscale, imageData.chromakey, fill.on, fill.type,
fill.color, fill.opacity, fill.color2, fill.method, fill.opacity2, fill.angle,
fill.focus, fill.focusposition.x, fill.focusposition.y, fill.focussize.x,
fill.focussize.y, stroke.on, stroke.color, stroke.weight, stroke.opacity,
stroke.linestyle, stroke.dashstyle, stroke.filltype, stroke.src, stroke.color2,
stroke.imagesize.x, stroke.imagesize.y, stroke.startArrow, stroke.endArrow,
stroke.startArrowWidth, stroke.startArrowLength, stroke.endArrowWidth,
stroke.endArrowLength, shadow.on, shadow.type, shadow.color, shadow.color2,
shadow.opacity, shadow.offset.x, shadow.offset.y, shadow.offset2.x,
shadow.offset2.y, shadow.origin.x, shadow.origin.y, shadow.matrix.xtox,
shadow.matrix.ytox, shadow.matrix.xtox, shadow.matrix.ytoy,
shadow.matrix.perspectiveX, shadow.matrix.perspectiveY, skew.on, skew.offset.x,
skew.offset.y, skew.origin.x, skew.origin.y, skew.matrix.xtox, skew.matrix.ytox,
skew.matrix.xtox, skew.matrix.ytoy, skew.matrix.perspectiveX,
skew.matrix.perspectiveY, extrusion.on, extrusion.type, extrusion.render,
extrusion.viewpointorigin.x, extrusion.viewpointorigin.y, extrusion.viewpoint.x,
extrusion.viewpoint.y, extrusion.viewpoint.z, extrusion.plane,
extrusion.skewangle, extrusion.skewamt, extrusion.backdepth,
extrusion.foredepth, extrusion.orientation.x, extrusion.orientation.y,
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
extrusion.orientation.z, extrusion.orientationangle, extrusion.color,
extrusion.rotationangle.x, extrusion.rotationangle.y,
extrusion.lockrotationcenter, extrusion.autorotationcenter,
extrusion.rotationcenter.x, extrusion.rotationcenter.y,
extrusion.rotationcenter.z, and extrusion.colormode.
```

[*Example*: Consider trying to emphasize the txt font size within the body of a shape. The attribute would be 'style.fontSize' and this can be done by doing the following:

```
<p:anim to="1.5" calcmode="lin" valueType="num">
  <p:cBhvr override="childStyle">
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:charRg st="4294967295" end="4294967295"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>style.fontSize</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:anim>
```

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

19.5.8 attrNameLst (Attribute Name List)

This element is used to describe a list of attributes in which to apply an animation to.

[*Example*: Consider trying to emphasize the txt font size within the body of a shape. The attribute would be 'style.fontSize' and this can be done by doing the following:

```
<p:anim to="1.5" calcmode="lin" valueType="num">
  <p:cBhvr override="childStyle">
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:charRg st="4294967295" end="4294967295"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <p:attrNameLst>
        <p:attrName>style.fontSize</p:attrName>
      </p:attrNameLst>
    </p:cBhvr>
  </p:anim>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLBehaviorAttributeNameList) is located in §A.3. *end note*]

19.5.9 audio (Audio)

This element is used to include audio during an animation. This element specifies that this node within the animation tree triggers the playback of an audio file; the actual audio file used is specified by the sndTgt element (§19.5.70).

[*Example*: Consider adding applause sound to an animation sequence. The audio element is used as follows:

```
<p:cTn …>
  <p:stCondLst>…</p:stCondLst>
  <p:childTnLst>…</p:childTnLst>
  <p:subTnLst>
    <p:audio>
      <p:cMediaNode vol="50%">…
        <p:tgtEl>
          <p:sndTgt r:embed="rId2" />
        </p:tgtEl>
      </p:cMediaNode>
    </p:audio>
  </p:subTnLst>
</p:cTn>
```

The audio element specifies the location of the audio playback within the animation; its child sndTgt element specifies that the audio to be played is the target of the relationship with ID `rId2`.

end example]

Attributes	Description
isNarration (Is Narration)	This attribute indicates whether the audio is a narration for the slide. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLMediaNodeAudio) is located in §A.3. *end note*]

19.5.10 bg (Background)

This element is used to specify animating the background of an object.

[*Example*: Consider adding animation to the background of Shape Id 3. The <bg> tag can be used as follows:

```
<p:tgtEl>
  <p:spTgt spid="3">
    <p:bg/>
  </p:spTgt>
</p:tgtEl>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.11 bldAsOne (Build As One)

This element specifies in the build list to build the entire graphical object as one entity.

[*Example*: Consider having a graph appear as on entity as opposed to by category. The <bldAsOne> element should be used as follows:

```
<p:bldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldAsOne/>
  </p:bldGraphic>
</p:bldLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.12 bldDgm (Build Diagram)

This element specifies how to build the animation for a diagram.

[*Example*: Consider the following example where a chart is specified to be animated by category rather than as one entity. Thus, the bldChart element should be used as follows:

```
<p:bdldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldSub>
      <a:bldChart bld="category"/>
    </p:bldSub>
  </p:bldGraphic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
</p:bldLst>
```

end example]

Attributes	Description
bld (Diagram Build Types)	This attribute describes how the diagram is built. The animation animates the sub-elements in the container in the particular order defined by this attribute. The possible values for this attribute are defined by the ST_TLDiagramBuildType simple type (§19.7.33).
grpId (Group ID)	This attribute ties effects persisted in the animation to the build information. The attribute is used by the editor when changes to the build information are made. GroupIDs are unique for a given shape. They are not guaranteed to be unique IDs across all shapes on a slide. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spid (Shape ID)	This attribute specifies the shape to which the build applies. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
uiExpand (Expand UI)	This attribute describes the view option indicating if the build should be displayed expanded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLBuildDiagram) is located in §A.3. *end note*]

19.5.13 bldGraphic (Build Graphics)

This element specifies how to build a graphical element.

[*Example*: Consider having a chart graphical element appear as a whole as opposed to by a category. The <bldGraphic> element should be used as follows:

```
<p:bldLdst>
  <p:bldGraphic spid="3" grpId="0">
    <p:bldSub>
      <a:bldChart bld="category"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bldLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
grpId (Group ID)	This attribute ties effects persisted in the animation to the build information. The attribute is used by the editor when changes to the build information are made. GroupIDs are unique for a given shape. They are not guaranteed to be unique IDs across all shapes on a slide. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spid (Shape ID)	This attribute specifies the shape to which the build applies. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
uiExpand (Expand UI)	This attribute describes the view option indicating if the build should be displayed expanded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLGraphicalObjectBuild) is located in §A.3. *end note*]

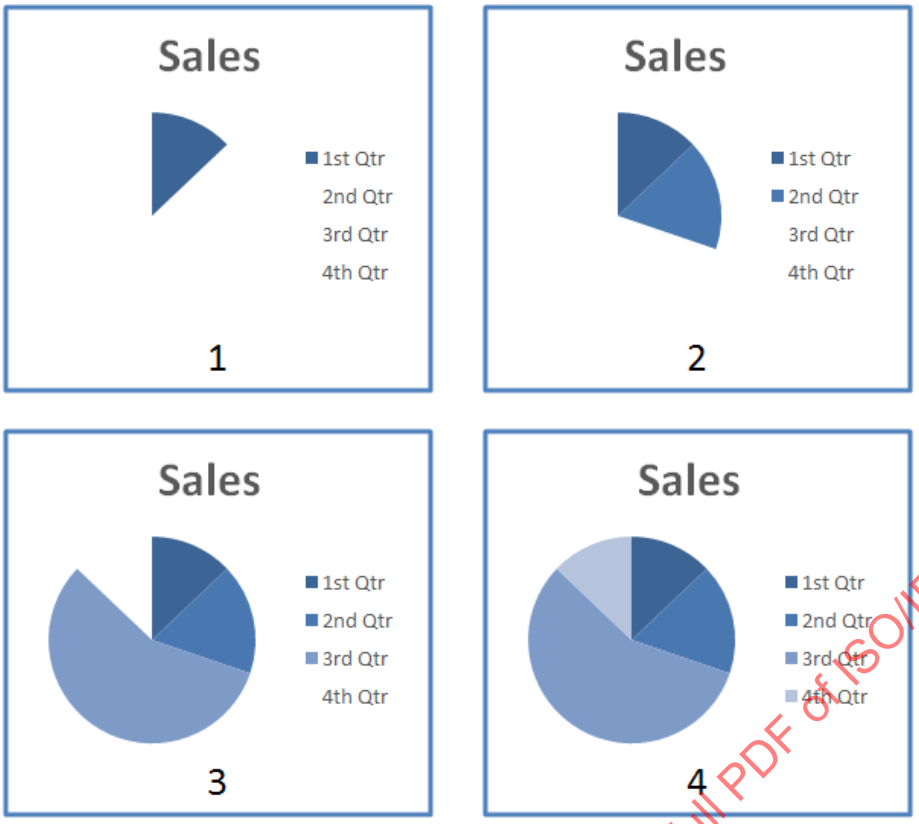
19.5.14 bldLst (Build List)

This element specifies the list of graphic elements to build. This refers to how the different sub-shapes or sub-components of a object are displayed. The different objects that can have build properties are text, diagrams, and charts.

[*Example*: Consider animating a pie chart but based on category as shown below:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



The <bldList> element should be used as follows:

```
<p:bldLst>
  <p:bldGraphic spid="1" grpId="0">
    <p:bldSub>
      <a:bldChart bld="category"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bldLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BuildList) is located in §A.3. *end note*]

19.5.15 bldOleChart (Build Embedded Chart)

This element describes animation an a embedded Chart.

[*Example*: Consider displaying animation on a embedded graphical chart. The <bldOleChart>element should be use as follows:

```
<p:bldLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <p:bldOleChart spid="1025" grpId="0"/>
  </p:bldLst>
```

end example]

Attributes	Description
animBg (Animate Background)	This attribute describes whether to animate the background of the shape. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bld (Build)	This attribute describes how the diagram is built. The animation animates the sub-elements in the container in the particular order defined by this attribute. The possible values for this attribute are defined by the ST_TLOleChartBuildType simple type (§19.7.35).
grpId (Group ID)	This attribute ties effects persisted in the animation to the build information. The attribute is used by the editor when changes to the build information are made. GroupIDs are unique for a given shape. They are not guaranteed to be unique IDs across all shapes on a slide. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spid (Shape ID)	This attribute specifies the shape to which the build applies. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
uiExpand (Expand UI)	This attribute describes the view option indicating if the build should be displayed expanded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLOleBuildChart) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.5.16 bldP (Build Paragraph)

This element specifies how to build paragraph level properties.

[*Example*: Consider having animation applied only to 1st level paragraphs. The <bldP> element should be used as follows:

```
  <p:bldLst>
    <p:bldP spid="3" grpId="0" build="p"/>
  </p:bldLst>
```

 ©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
advAuto (Auto Advance Time)	This attribute specifies time after which to automatically advance the build to the next step. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).
animBg (Animate Background)	This attribute indicates whether to animate the background of the shape associated with the text. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoUpdateAnimBg (Auto Update Animation Background)	This attribute indicates whether to automatically update the "animateBg" setting to true when the shape associated with the text has a fill or line. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bldLvl (Build Level)	This attribute describes the build level for the paragraph. It is only supported in paragraph type builds i.e the build attribute shall also be set to "byParagraph" for this attribute to apply. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
build (Build Types)	This attribute describe the build types. The possible values for this attribute are defined by the ST_TLParaBuildType simple type (§19.7.36).
grpId (Group ID)	This attribute ties effects persisted in the animation to the build information. The attribute is used by the editor when changes to the build information are made. GroupIDs are unique for a given shape. They are not guaranteed to be unique IDs across all shapes on a slide. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rev (Reverse)	This attribute is only supported in paragraph type builds. This specifies the direction of the build relative to the order of the elements in the container. When this is set to "true", the animations for the paragraphs are persisted in reverse order to the order of the paragraphs themselves such that the last paragraph animates first. Default value is "false". The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spid (Shape ID)	This attribute specifies the shape to which the build applies.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
uiExpand (Expand UI)	This attribute describes the view option indicating if the build should be displayed expanded. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLBuildParagraph) is located in §A.3. *end note*]

19.5.17 bldSub (Build Sub Elements)

This element specifies the animation properties of a graphical object's sub-elements.

[*Example*: Consider applying animation to a graphical element consisting of a diagram. The <bldSub> element should be used as follows:

```
<p:bldLst>
  <p:bldGraphic spid="5" grpId="0">
    <p:bldSub>
      <a:bldDgm bld="one"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bldLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationGraphicalObjectBuildProperties) is located in §A.4.1. *end note*]

19.5.18 blinds (Blinds Slide Transition)

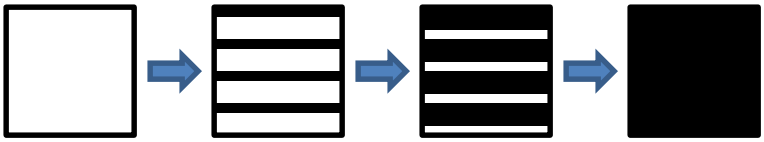
This element describes the blinds slide transition effect, which uses a set of horizontal or vertical bars and wipes them either left-to-right or top-to-bottom, respectively, until the new slide is fully shown. The rendering of this transition depends upon the attributes specified.

[*Example*: Consider the following cases in which the "blinds" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

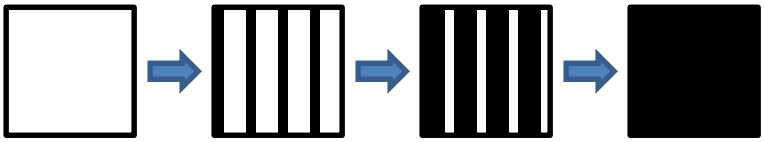
```
<p:transition>
  <p:blinds dir="horz"/>
</p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



```
<p:transition>
  <p:blinds dir="vert"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Transition Direction)	This attribute specifies a horizontal or vertical transition. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OrientationTransition) is located in §A.3. *end note*]

19.5.19 boolVal (Boolean Variant)

This element specifies a boolean value to be used for evaluation by a parent element. The exact meaning of the value contained within this element is not defined here but is dependent on the usage of this element in conjunction with one of the listed parent elements.

Attributes	Description
val (Value)	This attribute specifies the boolean value that this element contains and that should be used in evaluating this element. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariantBooleanVal) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.5.20 by (By)

This element describes the relative offset value for the color animation.

[*Example*: Consider a shape with a lightening emphasis animation applied to it. The <by> element should be used as follows:

```
<p:animClr clrSpc="hsl">
  <p:cBhvr>
    <p:cTn id="8" dur="500" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>stroke.color</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:by>
    <p:hsl h="0" s="0" l="0"/>
  </p:by>
</p:animClr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLByAnimateColorTransform) is located in §A.3. *end note*]

19.5.21 by (By)

This element describes the relative offset value for the animation.

[*Example*: Consider a shape with an animation effect that scales the size of an object by 150%. The <by> element should be used as follows:

```
<p:animScale>
  <p:cBhvr>
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
  </p:cBhvr>
  <p:by x="150.000%" y="150.000%"/>
</p:animScale>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
x (X coordinate)	This attribute describes the X coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
y (Y coordinate)	This attribute describes the Y coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLPoint) is located in §A.3. *end note*]

19.5.22 cBhvr (Common Behavior)

This element describes the common behaviors of animations.

[*Example*: Consider trying to emphasize text within a shape by changing the size of its font. The <anim> element should be used as follows:

```
<p:anim to="1.5" calcmode="lin" valueType="num">
  <p:cBhvr override="childStyle">
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:charRg st="4294967295" end="4294967295"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>style.fontSize</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:anim>
```

end example]

Attributes	Description
accumulate (Accumulate)	This attribute makes a repeating animation build with each iteration when set to "always." The possible values for this attribute are defined by the ST_TLBehaviorAccumulateType simple type (§19.7.27).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
additive (Additive)	This attribute specifies how to apply the animation values to the original value for the property. The possible values for this attribute are defined by the ST_TLBehaviorAdditiveType simple type (§19.7.28).
by (By)	This attribute specifies a relative offset value for the animation.. The possible values for this attribute are defined by the W3C XML Schema string datatype.
from (From)	This attribute specifies the starting value of the animation. The possible values for this attribute are defined by the W3C XML Schema string datatype.
override (Override)	This attribute specifies how a behavior should override values of the attribute being animated on the target element. The "childStyle" clears the attributes on the children contained inside the target element. The possible values for this attribute are defined by the ST_TLBehaviorOverrideType simple type (§19.7.29).
rctx (Runtime Context)	This attribute describes the runtime context of the animation. The currently-understood values are “PPT” and “IE.” This is used to specify the behavior used when animating in the PPT slideshow vs. IE HTML runtime. An example can be seen with the transparency effect. In IE, the transparency is animated as a bitmap, where in PPT, the style.opacity property of a shape is used to animate the transparency. The possible values for this attribute are defined by the W3C XML Schema string datatype.
to (To)	This attribute specifies the ending value of the animation. The possible values for this attribute are defined by the W3C XML Schema string datatype.
xfrmType (Transform Type)	This attribute specifies the kind of transform to be used. The possible values for this attribute are defined by the ST_TLBehaviorTransformType simple type (§19.7.30).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element’s content model (CT_TLCommonBehaviorData) is located in §A.3. *end note*]

19.5.23 charRg (Character Range)

This element specifies animation on a character range defined by a start and end character position.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider animating the first word (characters 1 through 9) within a sentence. The <charRg> element should be used as follows:

```
<p:animMotion>
  <p:cBhvr>
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:charRg st="0" end="9"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>ppt_x</p:attrName>
      <p:attrName>ppt_y</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
</p:animMotion>
```

end example]

Attributes	Description
end (End)	This attribute defines the end of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).
st (Start)	This attribute defines the start of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_IndexRange) is located in §A.3. *end note*]

19.5.24 checker (Checker Slide Transition)

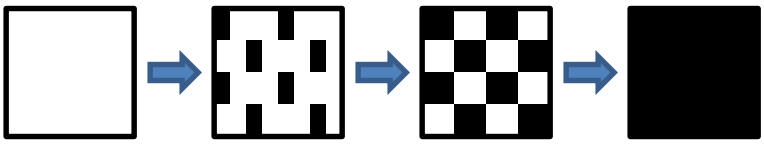
This element describes the checker slide transition effect, which uses a set of horizontal or vertical checkerboard squares and wipes them either left-to-right or top-to-bottom, respectively, until the new slide is fully shown. The rendering of this transition depends upon the attributes specified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

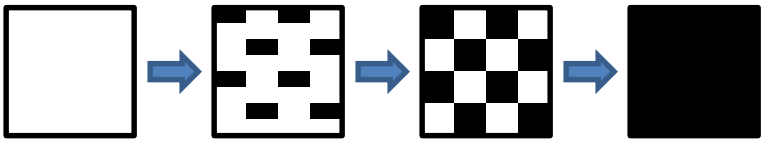
©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following cases in which the “checker” slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:checker dir="horz"/>
</p:transition>
```



```
<p:transition>
  <p:checker dir="vert"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Transition Direction)	This attribute specifies a horizontal or vertical transition. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).

[*Note*: The W3C XML Schema definition of this element’s content model (CT_OrientationTransition) is located in §A.3. *end note*]

19.5.25 childTnLst (Children Time Node List)

This element describes the list of time nodes that have a fixed location in the timing tree based on their parent time node. The children's start time is defined relative to their parent time node’s start.

[*Note*: The W3C XML Schema definition of this element’s content model (CT_TimeNodeList) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

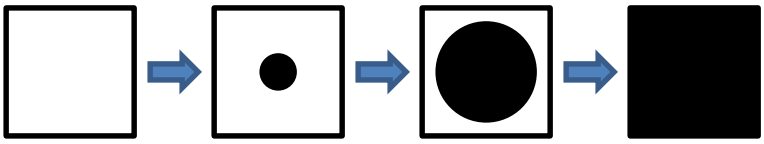
 ©ISO/IEC 2016 – All rights reserved

19.5.26 circle (Circle Slide Transition)

This element describes the circle slide transition effect, which uses a circle pattern centered on the slide that increases in size until the new slide is fully shown. The rendering of this transition has been shown below.

[*Example*: Consider the following case in which the “circle” slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:circle/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.3. *end note*]

19.5.27 clrVal (Color Value)

This element describes the color variant. This is used to specify a color that is to be used for animating the color property of an object.

[*Example*: Consider trying to emphasize text within a shape by changing the color its font.

```
<p:set>
  <p:cBhvr override="childStyle">
    …
  </p:cBhvr>
  <p:to>
    <p:clrVal>
      <a:schemeClr val="accent2"/>
    </p:clrVal>
  </p:to>
</p:set>
```

end example]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Color) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.28 cmd (Command)

This element describes the several non-durational commands that can be executed within a timeline. This can be used to send events, call functions on elements, and send verbs to embedded objects. For example “Object Action” effects for Embedded objects and Media commands for sounds/movies such as "PlayFrom(0.0)" and "togglePause".

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>cmd (Command)</td><td>
This attribute defines the actual command to be issued. Depending on the command specified, the actual command can be made to invoke a wide range of actions on the linked or embedded object.

Reserved Values (when type = “call”):
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>play</td><td>play corresponding media</td></tr>
<tr><td>playFrom(s)</td><td>play corresponding media starting from s, where s is the number of seconds from the beginning of the clip</td></tr>
<tr><td>pause</td><td>pause corresponding media</td></tr>
<tr><td>resume</td><td>resume play of corresponding media</td></tr>
<tr><td>stop</td><td>stop play of corresponding media</td></tr>
<tr><td>togglePause</td><td>play corresponding media if media is already paused, pause corresponding media if media is already playing. If the corresponding media is not active, this command restarts the media and plays from its beginning.</td></tr>
</table>

Reserved Values (when type = “evt”):
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>onstopaudio</td><td>stop play of all audio</td></tr>
</table>

Reserved Values (when type = “verb”):
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Open the object for editing</td></tr>
<tr><td>1</td><td>Open the object for viewing</td></tr>
</table>

The value of the cmd attribute shall be the string representation of an integer that represents the embedded object verb number. This verb number determines the action that the rendering application should take corresponding to this object when this point in the animation is reached.

[Example: Consider the following command
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<p:cmd type="evt" cmd="onstopaudio"> <p:cBhvr> <p:cTn display="0" masterRel="sameClick"> <p:stCondLst> <p:cond evt="begin" delay="0"> <p:tn val="5"/> </p:cond> </p:stCondLst> </p:cTn> <p:tgtEl> <p:sldTgt/> </p:tgtEl> </p:cBhvr> </p:cmd> *end example*] In the above example, the event of onstopaudio stops all audio from playing once this particular animation is reached in the timeline. The possible values for this attribute are defined by the W3C XML Schema string datatype.
type (Command Type)	This attribute specifies the kind of command that is issued by the rendering application to the appropriate target application or object. There are three possible values, `call`, `evt`, and `verb`. A `call` command type is used to specify the class of commands that can then be issued. Call commands (type="`call`"): This command type is used to call methods on the object specified (`play`(), `pause`(), etc.) Event Commands (type="`evt`"): This command type is used to set an event for the object at this point in the timeline (`onstopaudio`, etc.) Verb Commands (type="`verb`"): This command type is used to set verbs for the object to occur at this point in the timeline (0, 1, etc.) The possible values for this attribute are defined by the ST_TLCommandType simple type (§19.7.32).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLCommandBehavior) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.29 cMediaNode (Common Media Node Properties)

This element is used to describe behavior of media elements, such as sound or movies, in an animation.

[*Example*: Consider a shape with a sound effect attached to its animation. The <cMediaNode> element should be used as follows:

```
<p:audio>
  <p:cMediaNode mute="1">
    <p:cTn display="0" masterRel="sameClick">
      <p:stCondLst> … </p:stCondLst>
      <p:endCondLst> … </p:endCondLst>
    </p:cTn>
    <p:tgtEl> … </p:tgtEl>
  </p:cMediaNode>
</p:audio>
```

end example]

Attributes	Description
mute (Mute)	This attribute describes whether the media should be mute. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
numSld (Number of Slides)	This attribute describes the numbers of slides across which the media should play. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
showWhenStopped (Show When Stopped)	This attribute describes whether the media should be displayed when it is stopped. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vol (Volume)	This attribute describes the volume of the media element. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLCommonMediaNodeData) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

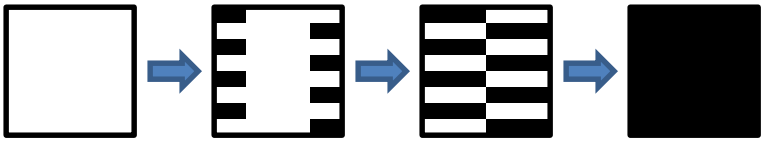
©ISO/IEC 2016 – All rights reserved

19.5.30 comb (Comb Slide Transition)

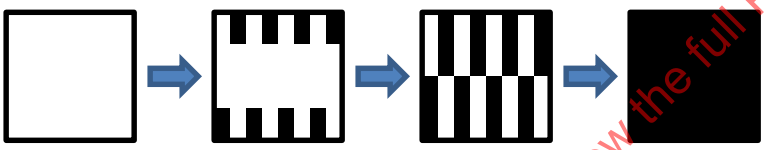
This element describes the comb slide transition effect, which uses a set of horizontal or vertical bars and wipes them from one end of the slide to the other until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "comb" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:comb dir="horz"/>
</p:transition>
```



```
<p:transition>
  <p:comb dir="vert"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Transition Direction)	This attribute specifies a horizontal or vertical transition. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OrientationTransition) is located in §A.3. *end note*]

19.5.31 cond (Condition)

This element specifies conditions on time nodes in a timeline. It is used within a list of start condition or list of end condition elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: For example, suppose we have a shape with a two second delay after the animation is started.

```
<p:cTn>
  <p:stCondLst>
    <p:cond delay="2000"/>
  </p:stCondLst>
  <p:childTnLst>
    <p:set> … </p:set>
    <p:animEffect transition="in" filter="blinds(horizontal)">
      <p:cBhvr>
        <p:cTn id="7" dur="1000"/>
        <p:tgtEl>
          <p:spTgt spid="4"/>
        </p:tgtEl>
      </p:cBhvr>
    </p:animEffect>
  </p:childTnLst>
</p:cTn>
```

end example]

Attributes	Description
delay (Trigger Delay)	This attribute describes the delay after an animation is triggered. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).
evt (Trigger Event)	This attribute describes the event that triggers an animation. The possible values for this attribute are defined by the ST_TLTriggerEvent simple type (§19.7.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeCondition) is located in §A.3. *end note*]

19.5.32 cover (Cover Slide Transition)

This element describes the cover slide transition effect, which moves the new slide in from an off-screen location, continually covering more of the previous slide until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

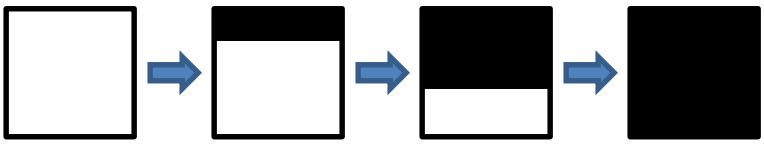
[*Example*: Consider the following cases in which the "cover" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
```

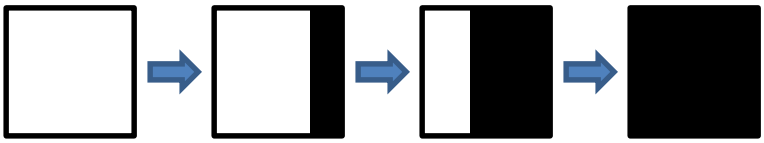
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

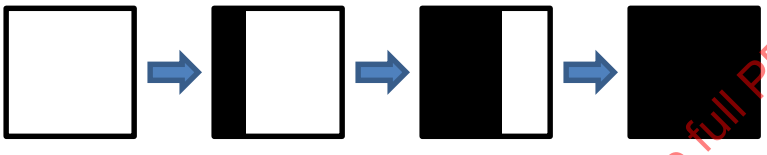
```
  <p:cover dir="d"/>
</p:transition>
```



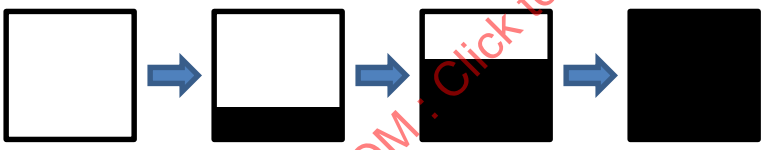
```
<p:transition>
  <p:cover dir="l"/>
</p:transition>
```



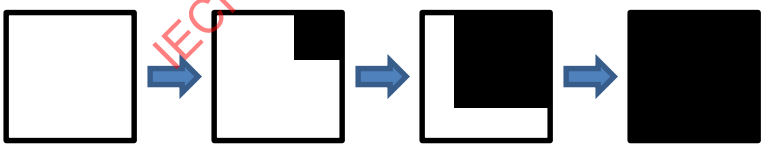
```
<p:transition>
  <p:cover dir="r"/>
</p:transition>
```



```
<p:transition>
  <p:cover dir="u"/>
</p:transition>
```



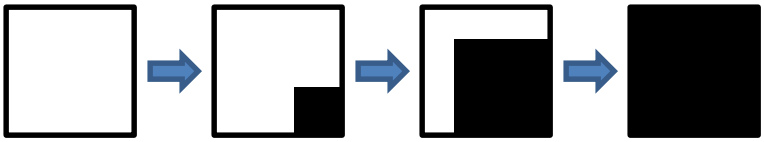
```
<p:transition>
  <p:cover dir="ld"/>
</p:transition>
```



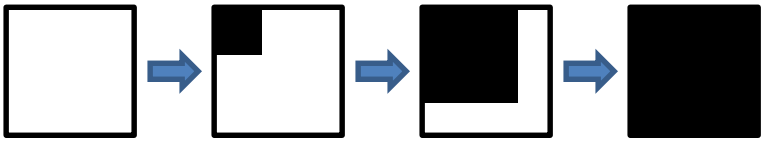
```
<p:transition>
  <p:cover dir="lu"/>
</p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

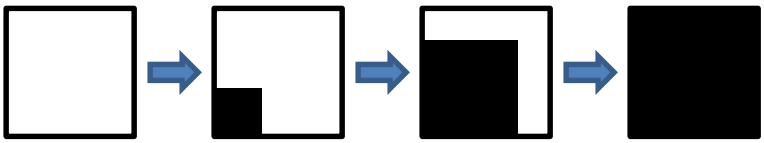
©ISO/IEC 2016 – All rights reserved



```
<p:transition>
  <p:cover dir="rd"/>
</p:transition>
```



```
<p:transition>
  <p:cover dir="ru"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies if the direction of the transition. The possible values for this attribute are defined by the ST_TransitionEightDirectionType simple type (§19.7.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EightDirectionTransition) is located in §A.3. *end note*]

19.5.33 cTn (Common Time Node Properties)

This element describes the properties that are common for time nodes.

Attributes	Description
accel (Acceleration)	This attribute describes the percentage of specified duration over which the element's time takes to accelerate from 0 up to the "run rate." The possible values for this attribute are defined by the ST_PositiveFixedPercentage

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	simple type (§20.1.10.45).
afterEffect (After Effect)	This attribute specifies whether there is an after effect applied to the time node. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
autoRev (Auto Reverse)	This attribute describes whether to automatically play the animation in reverse after playing it in the forward direction. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bldLvl (Build level)	This attribute describes the build level of the animation. The possible values for this attribute are defined by the W3C XML Schema int datatype.
decel (Deceleration)	This attribute describes the percentage of specified duration over which the element's time takes to decelerate from the "run rate" down to 0. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).
display (Display)	This attribute describes whether the state of the time node is visible or hidden. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
dur (Duration)	This attribute describes the duration of the time node, expressed as unit time. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).
evtFilter (Event Filter)	This attribute describes the event filter for this time node. The possible values for this attribute are defined by the W3C XML Schema string datatype.
fill (Fill)	This attribute describes the fill type for the time node. The possible values for this attribute are defined by the ST_TLTimeNodeFillType simple type (§19.7.41).
grpId (Group ID)	This attribute describes the Group ID of the time node. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
id (ID)	This attribute specifies the identifier for the timenode. The possible values for this attribute are defined by the ST_TLTimeNodeID simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§19.7.42).
masterRel (Master Relation)	This attribute specifies how the time node plays back relative to its master time node. The possible values for this attribute are defined by the ST_TLTimeNodeMasterRelation simple type (§19.7.43).
nodePh (Node Placeholder)	This attribute describes whether this node is a placeholder. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
nodeType (Node Type)	This attribute specifies the type of time node. The possible values for this attribute are defined by the ST_TLTimeNodeType simple type (§19.7.47).
presetClass (Preset Types)	This attribute descries the class of effect in which it belongs. The possible values for this attribute are defined by the ST_TLTimeNodePresetClassType simple type (§19.7.44).
presetID (Preset ID)	This attribute describes the preset identifier for the time node. The possible values for this attribute are defined by the W3C XML Schema int datatype.
presetSubtype (Preset SubType)	This attribute is a bitflag that specifies a direction or some other attribute of the effect. For example it can be set to specify a “From Bottom” for the Fly In effect, or “Bold” for the Change Font Style effect. The possible values for this attribute are defined by the W3C XML Schema int datatype.
repeatCount (Repeat Count)	This attribute describes the number of times the element should repeat, in units of thousandths. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).
repeatDur (Repeat Duration)	This attribute describes the amount of time over which the element should repeat. If absent, the attribute is taken to be the same as the specified duration. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).
restart (Restart)	This attribute specifies if a node is to restart when it completes its action. The possible values for this attribute are defined by the ST_TLTimeNodeRestartType simple type (§19.7.45).
spd (Speed)	This attribute specifies the percentage by which to speed up (or slow down) the timing. If negative, the timing is reversed. [*Example*: if speed is 200% and the specified duration is 10 seconds, the actual duration is 5 seconds. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
syncBehavior (Synchronization Behavior)	This attribute specifies how the time node synchronizes to its group. The possible values for this attribute are defined by the ST_TLTimeNodeSyncType simple type (§19.7.46).
tmFilter (Time Filter)	This attribute specifies the time filter for the time node. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLCommonTimeNodeData) is located in §A.3. *end note*]

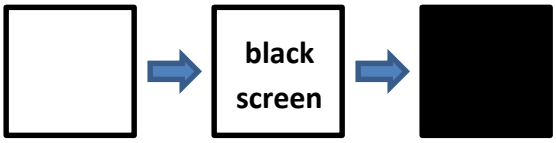
19.5.34 cut (Cut Slide Transition)

This element describes the cut slide transition effect, which simply replaces the previous slide with the new slide instantaneously. No animation is used, but an option exists to cut to a black screen before showing the new slide. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "cut" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:cut thruBlk="0"/>
</p:transition>
```

```
<p:transition>
  <p:cut thruBlk="1"/>
</p:transition>
```



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
thruBlk (Transition Through Black)	This attribute specifies if the transition starts from a black screen (and then transition the new slide over black). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

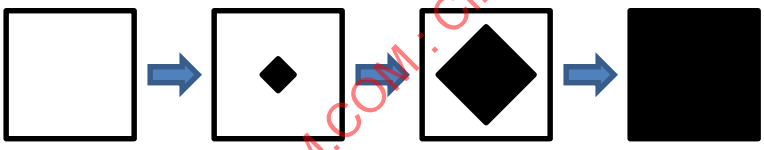
[*Note*: The W3C XML Schema definition of this element's content model (CT_OptionalBlackTransition) is located in §A.3. *end note*]

19.5.35 diamond (Diamond Slide Transition)

This element describes the diamond slide transition effect, which uses a diamond pattern centered on the slide that increases in size until the new slide is fully shown. The rendering of this transition has been shown below.

[*Example*: Consider the following case in which the "diamond" slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:diamond/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.36 dissolve (Dissolve Slide Transition)

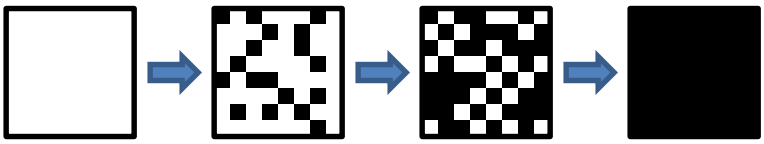
This element describes the dissolve slide transition effect, which uses a set of randomly placed squares on the slide that continue to be added to until the new slide is fully shown. The rendering of this transition has been shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following case in which the “dissolve” slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:dissolve/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.3. *end note*]

19.5.37 endCondLst (End Conditions List)

This element describes a list of the end conditions that shall be met in order to stop the time node.

[*Example*: Consider a shape a shape with an audio attached to the animation. The <endCondList> element should be used as follows to specifies when the sound is done:

```
<p:audio>
  <p:cMediaNode>
    <p:cTn display="0" masterRel="sameClick">
      <p:stCondLst> … </p:stCondLst>
      <p:endCondLst>
        <p:cond evt="onStopAudio" delay="0">
          <p:tgtEl>
            <p:sldTgt/>
          </p:tgtEl>
        </p:cond>
      </p:endCondLst>
    </p:cTn>
    <p:tgtEl> … </p:tgtEl>
  </p:cMediaNode>
</p:audio>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeConditionList) is located in §A.3. *end note*]

19.5.38 endSnd (Stop Sound Action)

This element stops all previous sounds during a slide transition.

[*Example*: Consider a slide transition that stops all previous sounds. The<endSnd> element should be used as follows:

```
<p:transition>
    <p:sndAc>
        <p:endSnd/>
    </p:sndAc>
</p:transition>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.39 endSync (EndSync)

This element is used to synchronizes the stopping of parallel elements in the timing tree. It is used on interactive timeline sequences to specify that the interactive sequence's duration ends when all of the child timenodes have ended. It is also used to make interactive sequences restart-able (so that the entire interactive sequence can be repeated if the trigger object is clicked on repeatedly).

[*Example*: Consider a shape with a fill change animation. The <endSync> element should be used as follows:

```
<p:seq concurrent="1" nextAc="seek">
  <p:cTn>
    <p:stCondLst/>
    <p:endSync evt="end" delay="0">
      <p:rtn val="all"/>
    </p:endSync>
    <p:childTnLst/>
  </p:cTn>
  <p:nextCondLst/>
</p:seq>
```

end example]

Attributes	Description
delay (Trigger Delay)	This attribute describes the delay after an animation is triggered. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

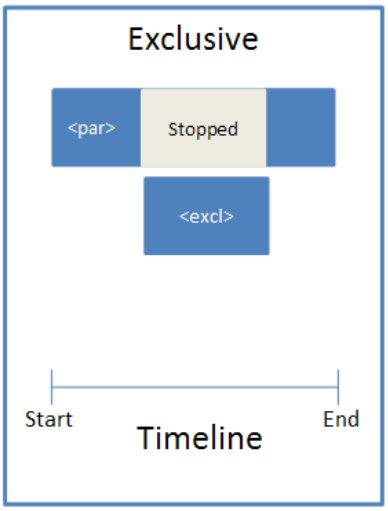
 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
evt (Trigger Event)	This attribute describes the event that triggers an animation. The possible values for this attribute are defined by the ST_TLTriggerEvent simple type (§19.7.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeCondition) is located in §A.3. *end note*]

19.5.40 excl (Exclusive)

This element describes the Exclusive time node. This time node is used to pause all other timelines when it is activated. Conceptually it can be though of as follows:



[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeNodeExclusive) is located in §A.3. *end note*]

19.5.41 fade (Fade Slide Transition)

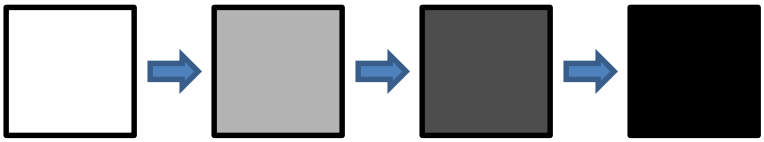
This element describes the fade slide transition effect, which smoothly fades the previous slide either directly to the new slide or first to a black screen and then to the new slide. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "fade" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

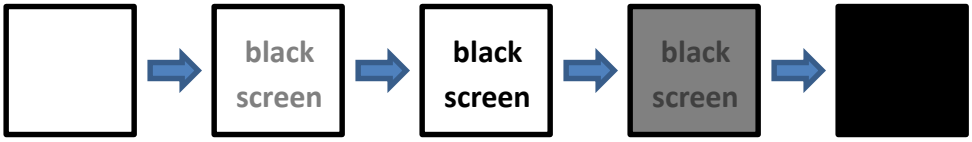
```
<p:transition>
  <p:fade thruBlk="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
</p:transition>
```



```
<p:transition>
  <p:fade thruBlk="1"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
thruBlk (Transition Through Black)	This attribute specifies if the transition starts from a black screen (and then transition the new slide over black). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OptionalBlackTransition) is located in §A.3. *end note*]

19.5.42 fltVal (Float Value)

This element specifies a floating point value to be used for evaluation by a parent element. The exact meaning of the value contained within this element is not defined here but is dependent on the usage of this element in conjunction with one of the listed parent elements.

Attributes	Description
val (Value)	This attribute specifies the floating point value that this element contains and that should be used in evaluating this element. The possible values for this attribute are defined by the W3C XML Schema float datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariantFloatVal) is located in §A.3. *end note*]

19.5.43 from (From)

This element is used to specify the starting color of the target element.

[*Example*: Consider a shape with an animation fill change from one accent color to another. The <from> element should be used as follows:

```
<p:animClr clrSpc="rgb" dir="cw">
  <p:cBhvr>
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl> … </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>fillcolor</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:from>
    <a:schemeClr val="accent3"/>
  </p:from>
  <p:to>
    <a:schemeClr val="accent2"/>
  </p:to>
</p:animClr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

19.5.44 from (From)

This element specifies an x/y co-ordinate to start the animation from.

[*Example*: Consider a shape with an animation sequence that needs to start at a certain coordinate. The <from> element should be used as follows:

```
<p:animScale>
  <p:cBhvr>
    <p:cTn> … </p:cTn>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
  </p:cBhvr>
  <p:from x="100%" y="100%"/>
  <p:to x="80%" y="100%"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</p:animScale>
```

end example]

Attributes	Description
x (X coordinate)	This attribute describes the X coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
y (Y coordinate)	This attribute describes the Y coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLPoint) is located in §A.3. *end note*]

19.5.45 graphicEl (Graphic Element)

This element specifies a graphical element which to animate.

[*Example*: Consider a diagram with an animation effect applied to it. The <graphicEl> element should be used as follows:

```
<p:set>
  <p:cBhvr>
    <p:cTn id="6" dur="1" fill="hold"> … </p:cTn>
    <p:tgtEl>
      <p:spTgt spid="4">
        <p:graphicEl>
          <a:dgm id="{87C2C707-C3F4-4E81-A967-A8B8AE13E575}"/>
        </p:graphicEl>
      </p:spTgt>
    </p:tgtEl>
    <p:attrNameLst> … </p:attrNameLst>
  </p:cBhvr>
  <p:to> … </p:to>
</p:set>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationElementChoice) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.5.46 hsl (HSL)

This element specifies an incremental HSL (Hue, Saturation, Lightness) value to add to a color animation.

[*Example*: Consider a shape with a lightening emphasis animation. The <hsl> element should be used as follows:

```
<p:animClr clrSpc="hsl">
  <p:cBhvr>
    <p:cTn id="8" dur="500" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>stroke.color</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:by>
    <p:hsl h="0" s="0" l="0"/>
  </p:by>
</p:animClr>
```

end example]

Attributes	Description
h (Hue)	Specifies hue as an angle. The values range from [0, 360] degrees. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).
l (Lightness)	Specifies a lightness as a percentage. The values are in the range [-100%, 100%]. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
s (Saturation)	Specifies a saturation as a percentage. The values are in the range [-100%, 100%]. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLByHslColorTransform) is located in §A.3. *end note*]

19.5.47 inkTgt (Ink Target)

This element specifies an animation target element that is represented by a sub-shape in a legacy graphical object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider an ink diagram with an animation blinds transition effect applied to it. The <inkTgt> element should be used as follows:

```
<p:animEffect transition="in" filter="blinds(horizontal)">
  <p:cBhvr>
    <p:cTn id="7" dur="500"/>
    <p:tgtEl>
      <p:inkTgt spid="_x0000_s2057"/>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

end example]

Attributes	Description
spid (Shape ID)	This attribute specifies the shape identifier. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLSubShapeId) is located in §A.3. *end note*]

19.5.48 intVal (Integer)

This element specifies an integer value to be used for evaluation by a parent element. The exact meaning of the value contained within this element is not defined here but is dependent on the usage of this element in conjunction with one of the listed parent elements.

Attributes	Description
val (Value)	This attribute specifies the integer value that this element contains and that should be used in evaluating this element. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariantIntegerVal) is located in §A.3. *end note*]

19.5.49 iterate (Iterate)

This element specifies how the animation should be successively applied to sub elements of the target element for a repeated effect. It can be applied to contained timing and animation structures over the letters, words, or shapes within a target element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a text animation where the words appear letter by letter. The <iterate> element should be used as follows:

```
<p:par>
  <p:cTn id="1" >
    <p:stCondLst> … </p:stCondLst>
    <p:iterate type="lt">
      <p:tmPct val="10000"/>
    </p:iterate>
    <p:childTnLst> … </p:childTnLst>
  </p:cTn>
</p:par>
```

end example]

Attributes	Description
backwards (Backwards)	This attribute specifies whether to go backwards in the timeline to the previous node. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
type (Iterate Type)	This attribute specifies the iteration behavior and applies it to each letter, word or shape within a container element. Values are by word, by letter, or by element. If there is no text or block elements such as shapes within the container or a single word, letter, or shape (depending on iterate type) then no iteration happens and the behavior is applied to the element itself instead. The possible values for this attribute are defined by the ST_IterateType simple type (§19.7.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLIterateData) is located in §A.3. *end note*]

19.5.50 newsflash (Newsflash Slide Transition)

This element describes the newsflash slide transition effect, which grows and spins the new slide counter-clockwise into place over the previous slide. The rendering of this transition has been shown below.

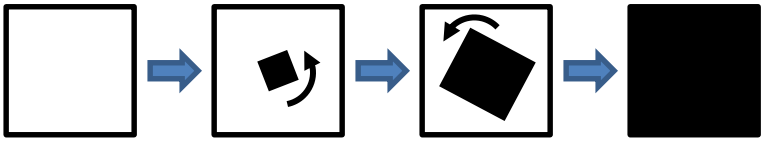
[*Example*: Consider the following case in which the "newsflash" slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <p:newsflash/>
  </p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.51 nextCondLst (Next Conditions List)

This element describes a list of conditions that shall be met to advance to the next animation sequence.

[*Example*: Consider a shape with a text emphasis changing the size of its font.

```
<p:seq concurrent="1" nextAc="seek">
  <p:cTn id="2" dur="indefinite" nodeType="mainSeq"> … </p:cTn>
  <p:prevCondLst> … </p:prevCondLst>
  <p:nextCondLst>
    <p:cond evt="onNext" delay="0">
      <p:tgtEl>
        <p:sldTgt/>
      </p:tgtEl>
    </p:cond>
  </p:nextCondLst>
</p:seq>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeConditionList) is located in §A.3. *end note*]

19.5.52 oleChartEl (Embedded Chart Element)

This element specifies the subelement of an embedded chart to animate.

[*Example*: Consider an embedded Chart with a entrance animation effect applied to each of the graph's categories. The <oldChartEl> element should be used as follows:

```
<p:animEffect transition="in" filter="blinds(horizontal)">
  <p:cBhvr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <p:cTn id="12" dur="500"/>
      <p:tgtEl>
        <p:spTgt spid="19460">
          <p:oleChartEl type="category" lvl="1"/>
        </p:spTgt>
      </p:tgtEl>
    </p:cBhvr>
  </p:animEffect>
```

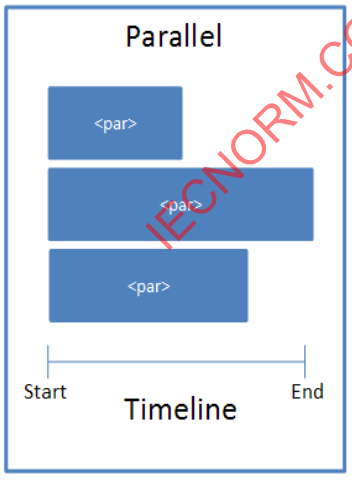
end example]

Attributes	Description
lvl (Level)	This attribute describes the element levels to animate. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Type)	This attribute specifies how to chart should be built during its animation. The possible values for this attribute are defined by the ST_TLChartSubelementType simple type (§19.7.31).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLOleChartTargetElement) is located in §A.3. *end note*]

19.5.53 par (Parallel Time Node)

This element describes the Parallel time node which can be activated along with other parallel time node containers. Conceptually it can be thought of as follows:



[*Example*: Consider a simple animation with a blind entrance. The <par> element should be used as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:timing>
  <p:tnLst>
    <p:par>
      <p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot">
        <p:childTnLst>
          <p:seq concurrent="1" nextAc="seek">
            …
          </p:seq>
        </p:childTnLst>
      </p:cTn>
    </p:par>
  </p:tnLst>
</p:timing>
```

end example]

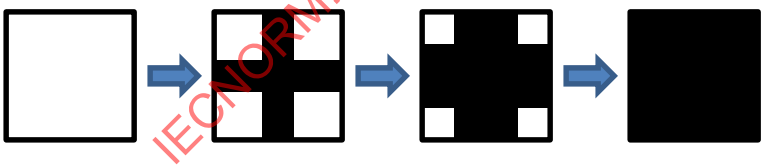
[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeNodeParallel) is located in §A.3. *end note*]

19.5.54 plus (Plus Slide Transition)

This element describes the plus slide transition effect, which uses a plus pattern centered on the slide that increases in size until the new slide is fully shown. The rendering of this transition has been shown below.

[*Example*: Consider the following case in which the "plus" slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:plus/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.55 prevCondLst (Previous Conditions List)

This element describes a list of conditions that shall be met in order to go backwards in an animation sequence.

[*Example*: Consider trying to emphasize text within a shape by changing the size of its font.

```
<p:seq concurrent="1" nextAc="seek">
  <p:cTn id="2" dur="indefinite" nodeType="mainSeq">
  </p:cTn>
  <p:prevCondLst>
    <p:cond evt="onPrev" delay="0">
      <p:tgtEl>
        <p:sldTgt/>
      </p:tgtEl>
    </p:cond>
  </p:prevCondLst>
  <p:nextCondLst>
  </p:nextCondLst>
</p:seq>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeConditionList) is located in §A.3. *end note*]

19.5.56 pRg (Paragraph Text Range)

This element specifies a text range to animate based on starting and ending paragraph number.

[*Example*: Consider an animation entrance of the first 3 text paragraphs. The <pRg> element should be used as follows:

```
<p:animEffect transition="in" filter="checkerboard(across)">
  <p:cBhvr>
    <p:cTn id="12" dur="500"/>
    <p:tgtEl>
      <p:spTgt spid="3">
        <p:txEl>
          <p:pRg st="0" end="2"/>
        </p:txEl>
      </p:spTgt>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
end (End)	This attribute defines the end of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).
st (Start)	This attribute defines the start of the index range. The possible values for this attribute are defined by the ST_Index simple type (§19.7.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_IndexRange) is located in §A.3. *end note*]

19.5.57 progress (Progress)

This element defines the progression of an animation. The default for the way animation progress happens through an animEffect is a linear ramp from 0 to 1, starting at the effect's begin time & ending at the effect's end time. When you specify a value for the progress attribute, you are overriding this default behaviour. The value between 0 and 1 represents a percentage through the effect, where 0 is 0% and 1 is 100%.

Each animEffect is in fact an object-based transition. These transitions can be specified as "In" (where the object is not visible at 0% and becomes completely visible at 100%) or "Out" (where the object is visible at 0% and becomes completely invisible at 100%). You would set the progress attribute if you want to use the animEffect as a "static" effect, where the transition properties do not actually change over time. As an alternative to using the progress attribute, you can use the tmFilter (time filter), which is a base attribute of any effect/timenode, to specify the way that progress through an effect should be performed dynamically.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariant) is located in §A.3. *end note*]

19.5.58 pull (Pull Slide Transition)

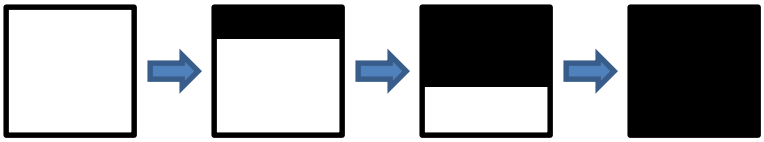
This element describes the pull slide transition effect, which moves the previous slide to an off-screen location, continually revealing more of the new slide until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "pull" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

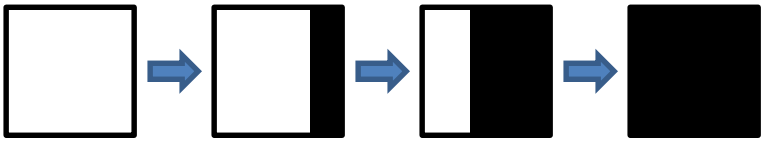
```
<p:transition>
  <p:pull dir="d"/>
</p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

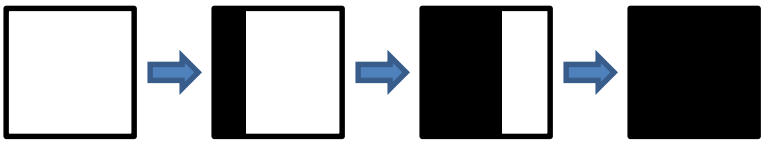
 ©ISO/IEC 2016 – All rights reserved



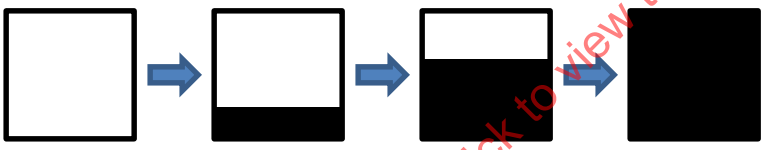
```
<p:transition>
  <p:pull dir="l"/>
</p:transition>
```



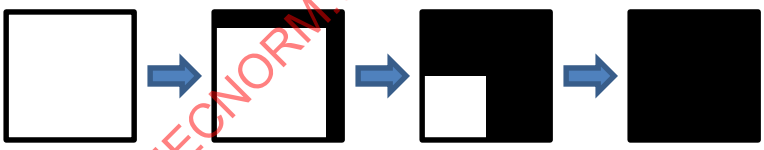
```
<p:transition>
  <p:pull dir="r"/>
</p:transition>
```



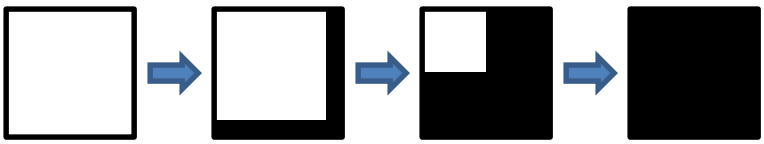
```
<p:transition>
  <p:pull dir="u"/>
</p:transition>
```



```
<p:transition>
  <p:pull dir="ld"/>
</p:transition>
```



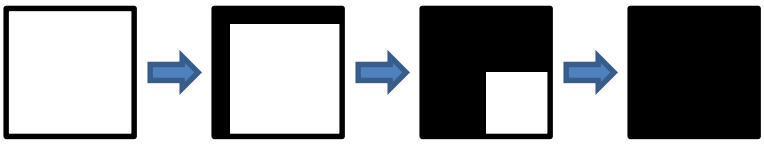
```
<p:transition>
  <p:pull dir="lu"/>
</p:transition>
```



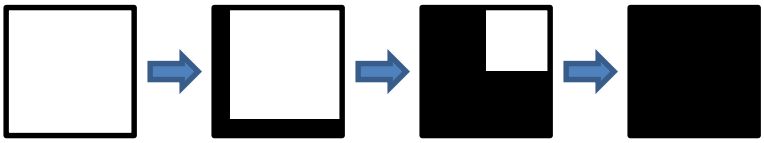
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:transition>
  <p:pull dir="rd"/>
</p:transition>
```



```
<p:transition>
  <p:pull dir="ru"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies if the direction of the transition. The possible values for this attribute are defined by the ST_TransitionEightDirectionType simple type (§19.7.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EightDirectionTransition) is located in §A.3. *end note*]

19.5.59 push (Push Slide Transition)

This element describes the push slide transition effect, which moves the new slide in from an off-screen location, continually pushing the previous slide to an opposite off-screen location until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

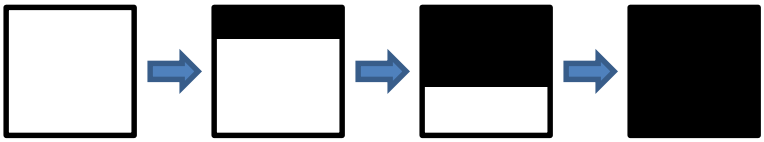
[*Example*: Consider the following cases in which the "push" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:push dir="d"/>
```

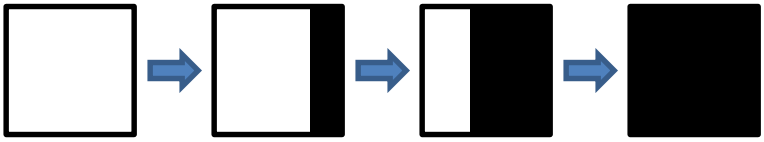
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

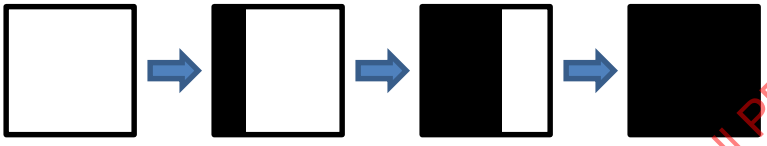
```
</p:transition>
```



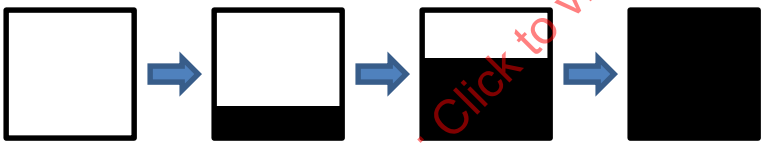
```
<p:transition>
  <p:push dir="l"/>
</p:transition>
```



```
<p:transition>
  <p:push dir="r"/>
</p:transition>
```



```
<p:transition>
  <p:push dir="u"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies the direction of the slide transition. The possible values for this attribute are defined by the ST_TransitionSideDirectionType simple type (§19.7.53).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SideDirectionTransition) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.5.60 random (Random Slide Transition)

This element describes the random slide transition effect, which chooses a random transition from the set available in the rendering application. This transition thus can be different each time it is used.

[*Example*: Consider the following case in which the “random” slide transition is applied to a slide, along with a set of attributes. The proper usage is shown below:

```
<p:transition>
  <p:random/>
</p:transition>
```

end example]

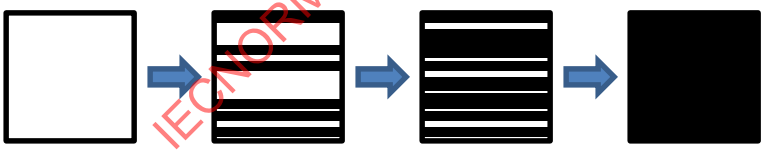
[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.3. *end note*]

19.5.61 randomBar (Random Bar Slide Transition)

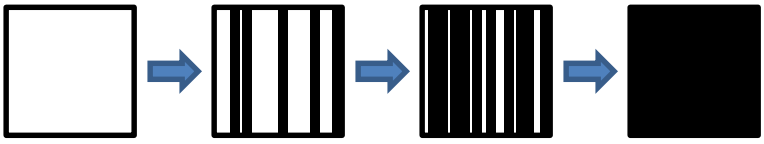
This element describes the randomBar slide transition effect, which uses a set of randomly placed horizontal or vertical bars on the slide that continue to be added to until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the “randomBar” slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:randomBar dir="horz"/>
</p:transition>
```



```
<p:transition>
  <p:randomBar dir="vert"/>
</p:transition>
```



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Transition Direction)	This attribute specifies a horizontal or vertical transition. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OrientationTransition) is located in §A.3. *end note*]

19.5.62 rCtr (Rotation Center)

This element describes the center of the rotation used to rotate a motion path by X angle.

[*Example*: For example, suppose we have a simple animation with a checkerbox text entrance.

```
<p:animMotion origin="layout" path="M 0 0 L 0.25 0.33333 E"
pathEditMode="relative" rAng="0" ptsTypes="">
  <p:cBhvr>
    <p:cTn id="6" dur="2000" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="3"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>ppt_x</p:attrName>
      <p:attrName>ppt_y</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:rCtr x="56.7%" y="83.4%"/>
</p:animMotion>
```

end example]

Attributes	Description
x (X coordinate)	This attribute describes the X coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
y (Y coordinate)	This attribute describes the Y coordinate.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLPoint) is located in §A.3. *end note*]

19.5.63 rgb (RGB)

The element specifies an incremental RGB value to add to the color property.

[*Example*: Consider a shape with a color emphasis animation. The <rgb> element should be used as follows:

```
<p:animClr clrSpc="rgb">
  <p:cBhvr>
    <p:cTn id="8" dur="500" fill="hold"/>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>stroke.color</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:by>
    <p:rgb r="10" g="20" b="30"/>
  </p:by>
</p:animClr>
```

end example]

Attributes	Description
b (Blue)	This attribute specifies a blue component luminance as a percentage. Values are in the range [-100%, 100%]. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
g (Green)	This attribute specifies a green component luminance as a percentage. Values are in the range [-100%, 100%]. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
r (Red)	This attribute specifies a red component luminance as a percentage. Values are in the range [-100%, 100%].

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLByRgbColorTransform) is located in §A.3. *end note*]

19.5.64 rtn (Runtime Node Trigger Choice)

This element specifies the child time node that triggers a time condition. References a child time node or all child nodes. Order is based on the child's end time.

[*Example*: Consider an animation which ends the synchronization of all parallel time nodes when all the child nodes have ended their animation. The <rtn> element should be used as follows:

```
<p:cTn>
  <p:stCondLst> … </p:stCondLst>
  <p:endSync evt="end" delay="0">
    <p:rtn val="all"/>
  </p:endSync>
  <p:childTnLst> … </p:childTnLst>
</p:cTn>
```

end example]

Attributes	Description
val (Value)	This attribute describes the value that triggers the runtime node. The possible values for this attribute are defined by the ST_TLTriggerRuntimeNode simple type (§19.7.49).

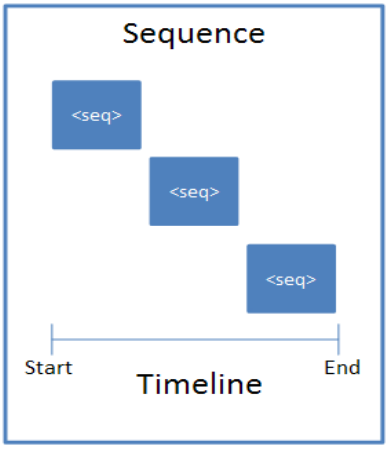
[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTriggerRuntimeNode) is located in §A.3. *end note*]

19.5.65 seq (Sequence Time Node)

This element describes the Sequence time node and it can only be activated when the one before it finishes. Conceptually it can be though of as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



[*Example*: For example, suppose we have a simple animation with a blind entrance.

```
<p:timing>
  <p:tnLst>
    <p:par>
      <p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot">
        <p:childTnLst>
          <p:seq concurrent="1" nextAc="seek">
            …
          </p:seq>
        </p:childTnLst>
      </p:cTn>
    </p:par>
  </p:tnLst>
</p:timing>
```

end example]

Attributes	Description
concurrent (Concurrent)	This attribute specifies if concurrency is enabled or disabled. By default this attribute has a value of "disabled". When the value is set to "enabled", the previous element is left enabled when advancing to the next element in a sequence instead of being ended. This is only relevant for advancing via the next condition element being triggered. The only other way to advance to the next element would be to have the current element end, which implies it is no longer concurrent. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
nextAc (Next Action)	This attribute specifies what to do when going forward in sequence. By default this attribute has a value of "none". When this is set to seek it seeks the element to a natural end time (not necessarily the actual end time).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The natural end position is defined as the latest non-infinite end time of the children. If a child loops forever, the end of its first loop is used as its "end time" for the purposes of this calculation. Some container elements can have infinite durations due to an infinite-duration child element. The engine needs to recurse down through all infinite duration containers to calculate their natural duration in case a child might have non-infinite duration within it that needs to be taken into account. The possible values for this attribute are defined by the ST_TLNextActionType simple type (§19.7.34).
prevAc (Previous Action)	This attribute specifies what to do when going backwards in a sequence. By default it is set to "none" and nothing special is done. When the value is "skipTimed", the sequence continues to go backwards until it reaches a sequence element that was defined to begin only on the next condition element. The possible values for this attribute are defined by the ST_TLPreviousActionType simple type (§19.7.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeNodeSequence) is located in §A.3. *end note*]

19.5.66 set (Set Time Node Behavior)

This element allows the setting of a particular property value to a fixed value while the behavior is active and restores the value when the behavior is reset or turned off.

[*Example*: For example, suppose we want to set certain properties during an animation effect. The <set> element should be used as follows:

```
<p:childTnLst>
  <p:set>
    <p:cBhvr>
      <p:cTn id="6" dur="1" fill="hold"> … </p:cTn>
      <p:tgtEl>
        <p:spTgt spid="4"/>
      </p:tgtEl>
      <p:attrNameLst>
        <p:attrName>style.visibility</p:attrName>
      </p:attrNameLst>
    </p:cBhvr>
    <p:to>
      <p:strVal val="visible"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      </p:to>
    </p:set>
    <p:animEffect transition="in" filter="blinds(horizontal)">
      …
    </p:animEffect>
  </p:childTnLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLSetBehavior) is located in §A.3. *end note*]

19.5.67 sldTgt (Slide Target)

This element specifies the slide as the target element.

[*Example*: For example, suppose we have a simple animation with a blind entrance.

```
  <p:seq concurrent="1" nextAc="seek">
    <p:cTn id="2" dur="indefinite" nodeType="mainSeq"> … </p:cTn>
    <p:prevCondLst> … </p:prevCondLst>
    <p:nextCondLst>
      <p:cond evt="onNext" delay="0">
        <p:tgtEl>
          <p:sldTgt/>
        </p:tgtEl>
      </p:cond>
    </p:nextCondLst>
  </p:seq>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.3. *end note*]

19.5.68 snd (Sound)

This element specifies the audio information to play during a slide transition.

[*Example*: Consider a slide transition with an audio effect. The <snd> element should be used as follows:

```
      <p:transition>
          <p:sndAc>
              <p:stSnd>
                  <p:snd r:embed="rId2""/>
              </p:stSnd>
          </p:sndAc>
      </p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

19.5.69 sndAc (Sound Action)

This element describes a sound action for slide transition. This element specifies that the start of the slide transition is accompanied by the playback of an audio file; the actual audio file used is specified by the snd element (§19.5.68).

[*Example*: Consider a slide transition with a sound effect. The <sndAc> element should be used as follows:

```
<p:transition>
    <p:sndAc>
        <p:stSnd>
            <p:snd r:embed="rId2"/>
        </p:stSnd>
    </p:sndAc>
</p:transition>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TransitionSoundAction) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.5.70 sndTgt (Sound Target)

This element describes the sound information for a target object.

[*Example*: Consider a shape with a sound effect animation. The <sndTgt> element should be used as follows:

```
<p:subTnLst>
  <p:audio>
    <p:cMediaNode>
      <p:cTn display="0" masterRel="sameClick"> … </p:cTn>
      <p:tgtEl>
        <p:sndTgt r:embed="rId2" r:link="rId3"/>
      </p:tgtEl>
    </p:cMediaNode>
  </p:audio>
</p:subTnLst>
```

end example]

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

19.5.71 split (Split Slide Transition)

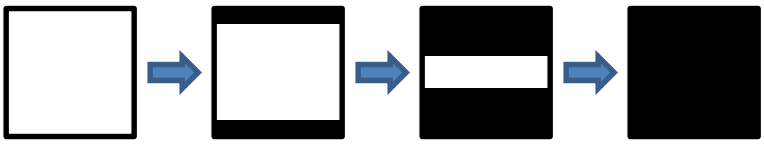
This element describes the split slide transition effect, which reveals the new slide directly on top of the previous one by wiping either horizontal or vertical from the outside in, or from the inside out, until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

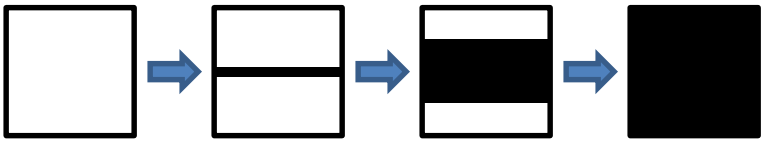
 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following cases in which the “split” slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

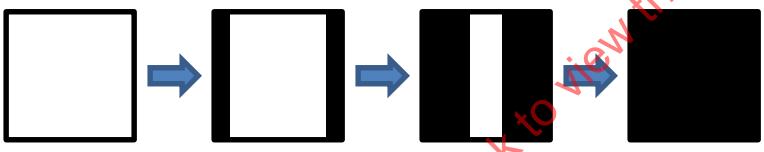
```
<p:transition>
  <p:split orient="horz" dir="in"/>
</p:transition>
```



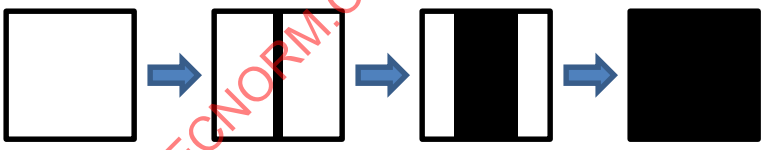
```
<p:transition>
  <p:split orient="horz" dir="out"/>
</p:transition>
```



```
<p:transition>
  <p:split orient="vert" dir="in"/>
</p:transition>
```



```
<p:transition>
  <p:split orient="vert" dir="out"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies the direction of a "split" slide transition.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TransitionInOutDirectionType simple type (§19.7.52).
orient (Orientation)	This attribute specifies the orientation of a "split" slide transition. The possible values for this attribute are defined by the ST_Direction simple type (§19.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SplitTransition) is located in §A.3. *end note*]

19.5.72 spTgt (Shape Target)

The element specifies the shape in which to apply a certain animation to.

[*Example*: Consider a shape whose id is 3 in which we want to apply a fade animation effect. The <spTgt> should be used as follows:

```
<p:animEffect transition="in" filter="fade">
  <p:cBhvr>
    <p:cTn id="7" dur="2000"/>
    <p:tgtEl>
      <p:spTgt spid="3"/>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

end example]

Attributes	Description
spid (Shape ID)	This attribute specifies the shape identifier. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLShapeTargetElement) is located in §A.3. *end note*]

19.5.73 stCondLst (Start Conditions List)

This element contains a list conditions that shall be met for a time node to be activated.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: For example, suppose we have a shape with an entrance appearance after 5 seconds. The <stCondLst>element should be used as follows:

```
<p:par>
  <p:cTn id="5" nodeType="clickEffect">
    <p:stCondLst>
      <p:cond delay="5000"/>
    </p:stCondLst>
    <p:childTnLst> … </p:childTnLst>
  </p:cTn>
</p:par>
```

end example]

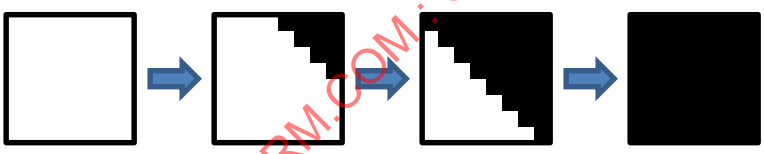
[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeConditionList) is located in §A.3. *end note*]

19.5.74 strips (Strips Slide Transition)

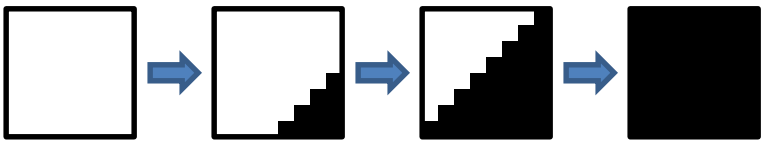
This element describes the strips slide transition effect, which uses a set of bars that are arranged in a staggered fashion and wipes them across the screen until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "strips" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:strips dir="ld"/>
</p:transition>
```



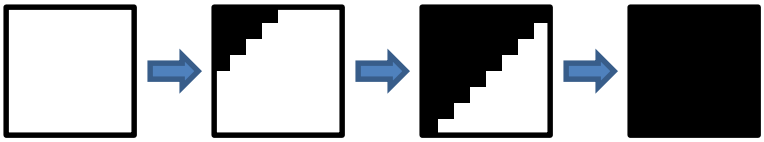
```
<p:transition>
  <p:strips dir="lu"/>
</p:transition>
```



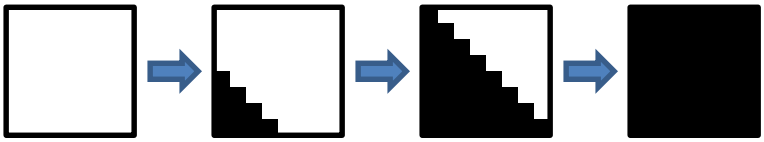
```
<p:transition>
  <p:strips dir="rd"/>
</p:transition>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



```
<p:transition>
  <p:strips dir="ru"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies if the direction of the transition. The possible values for this attribute are defined by the ST_TransitionCornerDirectionType simple type (§19.7.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CornerDirectionTransition) is located in §A.3. *end note*]

19.5.75 strVal (String Value)

This element specifies a string value to be used for evaluation by a parent element. The exact meaning of the value contained within this element is not defined here but is dependent on the usage of this element in conjunction with one of the listed parent elements.

Attributes	Description
val (Value)	This attribute specifies the string value that this element contains and that should be used in evaluating this element. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariantStringVal) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.5.76 stSnd (Start Sound Action)

This element describes the sound that starts playing during a slide transition.

[*Example*: Consider a slide transition that starts with a sound effect. The <stSnd> element should be used as follows:

```
<p:transition>
    <p:sndAc>
        <p:stSnd>
            <p:snd r:embed="rId2"/>
        </p:stSnd>
    </p:sndAc>
</p:transition>
```

end example]

Attributes	Description
loop (Loop Sound)	This attribute specifies if the sound loops until the next sound event occurs in slideshow. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TransitionStartSoundAction) is located in §A.3. *end note*]

19.5.77 subSp (Subshape)

This element specifies the subshape of a legacy graphical object to animate.

[*Example*: Consider adding animation to a legacy diagram. The <subSp> element should be used as follows:

```
<p:animEffect transition="in" filter="blinds(horizontal)">
  <p:cBhvr>
    <p:cTn id="7" dur="500"/>
    <p:tgtEl>
      <p:spTgt spid="2053">
        <p:subSp spid="_x0000_s70664"/>
      </p:spTgt>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
spid (Shape ID)	This attribute specifies the shape identifier. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLSubShapeId) is located in §A.3. *end note*]

19.5.78 subTnLst (Sub-TimeNodes List)

This element describes time nodes that have a start time which is not based on the containing timenode. It is instead based on their master relationship (masterRel). At runtime, they are inserted dynamically into the timing tree as child timenodes for playback, based on the logic defined by the master relationship. These elements are used for animations such as "dim after" and "play sound effects".

[*Example*: Consider an animation with a "Fly In" effect on paragraphs so that each paragraph flies in on a separate click. Then the "Dim After" effect for paragraph 1 doe not happen until paragraph 2 flies in. The <subTnLst> element should be used as follows:

```
<p:par>
  <p:cTn id="5" grpId="0" nodeType="clickEffect">
    <p:stCondLst> … </p:stCondLst>
    <p:childTnLst> … </p:childTnLst>
    <p:subTnLst>
      <p:set>
        <p:cBhvr override="childStyle">
          <p:cTn fill="hold" masterRel="nextClick" afterEffect="1"/>
          <p:tgtEl> … </p:tgtEl>
          <p:attrNameLst> … </p:attrNameLst>
        </p:cBhvr>
        <p:to> … </p:to>
      </p:set>
    </p:subTnLst>
  </p:cTn>
</p:par>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeNodeList) is located in §A.3. *end note*]

19.5.79 tav (Time Animate Value)

This element defines a "keypoint" in animation interpolation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a shape with a "fly-in" animation. The <tav> element should be used as follows:

```
<p:anim calcmode="lin" valueType="num">
  <p:cBhvr additive="base"> … </p:cBhvr>
  <p:tavLst>
    <p:tav tm="0%">
      <p:val>
        <p:strVal val="1+#ppt_h/2"/>
      </p:val>
    </p:tav>
    <p:tav tm="100%">
      <p:val>
        <p:strVal val="#ppt_y"/>
      </p:val>
    </p:tav>
  </p:tavLst>
</p:anim>
```

end example]

Attributes	Description
fmla (Formula)	This attribute allows for the specification of a formula to be used for describing a complex motion for an animated object. The formula manipulates the motion of the object by modifying a property of the object over a specified period of time. Each formula has zero or more inputs specified by the ($) symbol, zero or more variables specified by the (#) symbol pre-pended to the variable name and a target variable which is specified by the previously specified attrName element. The formula can contain one or more of any of the constants, operators or functions listed below. In addition to this, the formula can also contain floating point numbers and parentheses. Mathematical operations have the following order of precedence, listed from lowest to highest. Operators listed on the same line have equal precedence. • “+”, “-“ • “*”, “/”, “%” • “^” • Unary minus, Unary plus (e.g. -2, meaning 3*-2 is the same as 3*(-2)) • Variables, Constants (including numbers) and Functions (as listed previously) Language Description: `Digit = '0' | '1' | ‘2’ | ‘3’ | ‘4’ | ‘5’ | ‘6’ | ‘7’ | ‘8’ | '9' ;`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

```
number   = digit , { digit } ;
exponent = [ '-' ] , ( 'e' | 'E' ) , number ;
value    = number , [ '.' number ] , [ exponent ] ;
variable = '$' | 'ppt_x' | 'ppt_y' | 'ppt_w' | 'ppt_h' ;
constant = value | 'pi' | 'e' ;
ident    = 'abs' | 'acos' | 'asin' | 'atan' | 'ceil'
   | 'cos' | 'cosh' | 'deg' | 'exp' | 'floor' | 'ln'
   | 'max' | 'min' | 'rad' | 'rand' | 'sin' | 'sinh'
   | 'sqrt' | 'tan' | 'tanh' ;
function = ident , '(' , formula [ ',' , formula ] , ')' ;
formula  = term , { [ '+' | '-' ] , term } ;
term     = power , { [ '*' | '/' | '%' ] , power } ;
power    = unary [ '^' , unary ] ;
unary    = [ '+' | '-' ] , factor ;
factor   = variable | constant | function | parens ;
parens   = '(' , formula , ')' ;
```

[*Note:* Formulas can only support a calcMode (Calculation Mode) of linear or discrete. If another calcMode is specified or no calcMode is specified then a calcMode of linear is assumed. *end note*]

[*Note:* Any additional characters in the formula string that are not contained within the set described are considered invalid. *end note*]

Variables:

Name	Description
$	Formula input
ppt_x	Pre-animation x position of the object on the slide Position on the slide is represented using an abstract coordinate space where 0 represents the left side of the slide and 1 represents the right side of the slide.
ppt_y	Pre-animation y position of the object on the slide Position on the slide is represented using an abstract coordinate space where 0 represents the top of the slide and 1 represents the bottom of the slide.
ppt_w	Pre-animation width of the object Width is represented using an abstract coordinate space relative to the size of the slide, where 1 represents the width of the slide.
ppt_h	Pre-animation height of the object Height is represented using an abstract coordinate space relative to the size of the slide, where 1 represents the height of the slide.

Constants:

Name	Description

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>pi</td><td>Mathematical constant pi</td></tr>
<tr><td>e</td><td>Mathematical constant e</td></tr>
</table>
Operators:
<table>
<tr><th>Name</th><th>Description</th><th>Usage</th></tr>
<tr><td>+</td><td>Addition</td><td>“x+y”, adds x to the value y</td></tr>
<tr><td>-</td><td>Subtraction</td><td>“x-y”, subtracts y from the value x</td></tr>
<tr><td>*</td><td>Multiplication</td><td>“x*y”, multiplies x by the value y</td></tr>
<tr><td>/</td><td>Division</td><td>“x/y”, divides x by the value y</td></tr>
<tr><td>%</td><td>Modulus</td><td>“x%y”, the remainder of x/y</td></tr>
<tr><td>^</td><td>Power</td><td>“x^y”, x raised to the power y</td></tr>
</table>
Functions:
<table>
<tr><th>Name</th><th>Description</th><th>Usage</th></tr>
<tr><td>abs</td><td>Absolute value</td><td>“abs(x)”, absolute value of x</td></tr>
<tr><td>acos</td><td>Arc Cosine</td><td>“acos(x)”, arc cosine of the value x</td></tr>
<tr><td>asin</td><td>Arc Sine</td><td>“asin(x)”, arc sine of the value x</td></tr>
<tr><td>atan</td><td>Arc Tangent</td><td>“atan(x)”, arc tangent of the value x</td></tr>
<tr><td>ceil</td><td>Ceil value</td><td>“ceil(x)”, value of x rounded up</td></tr>
<tr><td>cos</td><td>Cosine</td><td>“cos(x)”, cosine of the value of x</td></tr>
<tr><td>cosh</td><td>Hyperbolic Cosine</td><td>“cosh(x)", hyperbolic cosine of the value x</td></tr>
<tr><td>deg</td><td>Radiant to Degree convert</td><td>“deg(x)”, the degree value of radiant value x</td></tr>
<tr><td>exp</td><td>Exponent</td><td>“exp(x)”, value of constant e raised to the power of x</td></tr>
<tr><td>floor</td><td>Floor value</td><td>“floor(x)”, value of x rounded down</td></tr>
<tr><td>ln</td><td>Natural logarithm</td><td>“ln(x)”, natural logarithm of x</td></tr>
<tr><td>max</td><td>Maximum of two values</td><td>“max(x,y)”, returns x if (x > y) or returns y if (y > x)</td></tr>
<tr><td>min</td><td>Minimum of two values</td><td>“min(x,y)", returns x if (x < y) or returns y if (y < x)</td></tr>
<tr><td>rad</td><td>Degree to Radiant convert</td><td>“rad(x)”, the radiant value of degree value x</td></tr>
<tr><td>rand</td><td>Random value</td><td>“rand(x)”, returns a random floating point value between 0 and x</td></tr>
<tr><td>sin</td><td>Sine</td><td>“sin(x)”, sine of the value x</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

sinh	Hyperbolic Sine	"sinh(x)", hyperbolic sine of the value x
sqrt	Square root	"sqrt(x)", square root of the value x
tan	Tangent	"tan(x)", tangent of the value x
tanh	Hyperbolic Tangent	"tanh(x)", hyperbolic tangent of the value x

[*Example:* Consider the following animation path:

```
<p:animcalcmode="lin" valueType="num">
  <p:cBhvr>
    <p:cTn id="9" dur="664" tmFilter="0.0,0.0; 0.25,0.07;
0.50,0.2; 0.75,0.467; 1.0,1.0">
      <p:stCondLst>
        <p:cond delay="0"/>
      </p:stCondLst>
    </p:cTn>
    <p:tgtEl>
      <p:spTgtspid="4"/>
    </p:tgtEl>
    <p:attrNameLst>
      <p:attrName>ppt_y</p:attrName>
    </p:attrNameLst>
  </p:cBhvr>
  <p:tavLst>
    <p:tav tm="0%" fmla="#ppt_y-sin(pi*$)/3">
      <p:val>
        <p:fltValval="0.5"/>
      </p:val>
    </p:tav>
    <p:tav tm="100%">
      <p:val>
        <p:fltValval="1"/>
      </p:val>
    </p:tav>
  </p:tavLst>
</p:anim>
```

The animation example above modifies the ppt_y variable of the object by subtracting sin(pi*$)/3 from the non-animated value of ppt_y. The start value is 0.5 and the end value is 1 specified in each of the val elements. The total time for this animation is specified within the dur attribute and the filtered time graph is specified by the tmFilter attribute. The end result is that the object moves from a point above its non-animated position back to its non-animated position. With the specification of the tmFilter it has a modified time graph such that it also appears to accelerate as it reaches its final position.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Note*: For this example, the non-animated value of ppt_y is the value of this variable if the object were to be statically rendered on the slide without animation properties. *end note*] *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
tm (Time)	This attribute specifies the time at which the attribute being animated takes on the value. The possible values for this attribute are defined by the ST_TLTimeAnimateValueTime simple type (§19.7.39).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeAnimateValue) is located in §A.3. *end note*]

19.5.80 tavLst (Time Animated Value List)

This element specifies a list of time animated value elements.

[*Example*: Consider a shape with a "fly-in" animation. The <tav> element should be used as follows:

```
<p:anim calcmode="lin" valueType="num">
  <p:cBhvr additive="base"> … </p:cBhvr>
  <p:tavLst>
    <p:tav tm="0%">
      <p:val>
        <p:strVal val="1+#ppt_h/2"/>
      </p:val>
    </p:tav>
    <p:tav tm="100000">
      <p:val>
        <p:strVal val="#ppt_y"/>
      </p:val>
    </p:tav>
  </p:tavLst>
</p:anim>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeAnimateValueList) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.81 tgtEl (Target Element)

This element specifies the target children elements which have the animation effects applied to.

[*Example*: Consider a shape with ID 3 with a fade effect animation applied to it. The <tgtEl> element should be used as follows:

```
<p:animEffect transition="in" filter="fade">
  <p:cBhvr>
    <p:cTn id="7" dur="2000"/>
    <p:tgtEl>
      <p:spTgt spid="3"/>
    </p:tgtEl>
  </p:cBhvr>
</p:animEffect>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTimeTargetElement) is located in §A.3. *end note*]

19.5.82 tmAbs (Time Absolute)

This element describes the duration of the iteration interval in absolute time.

[*Example*: Consider a text animation where the words appear letter by letter every 10 seconds. The <tmAbs> element should be used as follows:

```
<p:par>
  <p:cTn id="5" >
    <p:stCondLst> … </p:stCondLst>
    <p:iterate type="lt">
      <p:tmAbs val="10000"/>
    </p:iterate>
    <p:childTnLst> … </p:childTnLst>
  </p:cTn>
</p:par>
```

end example]

Attributes	Description
val (Time)	This attribute describes an amount of time, in milliseconds. The possible values for this attribute are defined by the ST_TLTime simple type (§19.7.38).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLIterateIntervalTime) is located in §A.3. *end note*]

19.5.83 tmPct (Time Percentage)

This element describes the duration of the iteration interval in a percentage of time.

[*Example*: Consider a text animation where the words appear letter by letter every 10th of the animation duration. The <tmPct> element should be used as follows:

```
<p:par>
  <p:cTn id="5" >
    <p:stCondLst> … </p:stCondLst>
    <p:iterate type="lt">
      <p:tmPct val="10%"/>
    </p:iterate>
    <p:childTnLst> … </p:childTnLst>
  </p:cTn>
</p:par>
```

end example]

Attributes	Description
val (Value)	This attribute specifies the time expressed as a percentage. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLIterateIntervalPercentage) is located in §A.3. *end note*]

19.5.84 tmpl (Template Effects)

This element specifies the "template" effects that are used by the build element. Template effects are used in text builds on the master slide. They define the rules of what effect should be applied to the 1st level paragraph, 2nd level paragraph, etc.

[*Example*: Consider a template with a fade in effect applied to it. The <tmpl> element should be used as follows:

```
<p:timing>
  <p:tnLst> … </p:tnLst>
  <p:bldLst>
    <p:bldP spid="3" grpId="0" build="p">
      <p:tmplLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
          <p:tmpl lvl="1">
          </p:tmpl>
        </p:tmplLst>
      </p:bldP>
    </p:bldLst>
  </p:timing>
```

end example]

Attributes	Description
lvl (Level)	This attribute describes the paragraph indent level to which this template effect applies. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTemplate) is located in §A.3. *end note*]

19.5.85 tmplLst (Template effects)

This element describes a list of template effects that describe what kind of effects should be applied to a paragraph level properties.

[*Example*: Consider a template with a fade in effect applied to it. The <tmpl> element should be used as follows:

```
<p:timing>
  <p:tnLst> … </p:tnLst>
  <p:bldLst>
    <p:bldP spid="3" grpId="0" build="p">
      <p:tmplLst>
        <p:tmpl lvl="1">
        </p:tmpl>
      </p:tmplLst>
    </p:bldP>
  </p:bldLst>
</p:timing>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTemplateList) is located in §A.3. *end note*]

19.5.86 tn (Time Node)

This element describes the time node trigger choice.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a time node with an event condition. The <tn> element should be used as follows:

```
<p:par>
  <p:cTn id="5">
    <p:stCondLst>
      <p:cond delay="0"/>
    </p:stCondLst>
    <p:endCondLst>
      <p:cond evt="begin" delay="0">
        <p:tn val="5"/>
      </p:cond>
    </p:endCondLst>
    <p:childTnLst> … </p:childTnLst>
  </p:cTn>
</p:par>
```

end example]

Attributes	Description
val (Value)	This attribute specifies a time node identifier. The possible values for this attribute are defined by the ST_TLTimeNodeID simple type (§19.7.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTriggerTimeNodeID) is located in §A.3. *end note*]

19.5.87 tnLst (Time Node List)

This element specifies a list of time node elements used in an animation sequence.

[*Example*: Consider a simple animation sequence. The <tnLst> element should be used as follows:

```
<p:timing>
  <p:tnLst>
    <p:par> … </p:par>
  </p:tnLst>
</p:timing>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeNodeList) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.5.88 to (To)

This element specifies the target location for an animation motion or animation scale effect

[*Example*: Consider an animation with a "light speed" entrance effect.

```
<p:animScale>
  <p:cBhvr>
    <p:cTn id="9" dur="200" decel="10.5%" autoRev="1" fill="hold">
      <p:stCondLst>
        <p:cond delay="600"/>
      </p:stCondLst>
    </p:cTn>
    <p:tgtEl>
      <p:spTgt spid="4"/>
    </p:tgtEl>
  </p:cBhvr>
  <p:from x="100%" y="100%"/>
  <p:to x="80%" y="100%"/>
</p:animScale>
```

end example]

Attributes	Description
x (X coordinate)	This attribute describes the X coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
y (Y coordinate)	This attribute describes the Y coordinate. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLPoint) is located in §A.3. *end note*]

19.5.89 to (To)

The element specifies the certain attribute of a time node after an animation effect.

[*Example*: Consider an animation effect that leaves a string value visible afterwards. The <to> element should be used as follows:

```
<p:childTnLst>
  <p:set>
    <p:cBhvr> … </p:cBhvr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <p:to>
        <p:strVal val="visible"/>
      </p:to>
    </p:set>
    <p:anim calcmode="lin" valueType="num"> … </p:anim>
  …</p:childTnLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariant) is located in §A.3. *end note*]

19.5.90 to (To)

This element specifies the resulting color for the animation color change.

[*Example*: Consider emphasize a shape by changing its fill color from blue to red. The <to> element should be used as follows:

```
<p:childTnLst>
  <p:animClr clrSpc="rgb">
    <p:cBhvr> … </p:cBhvr>
    <p:to>
      <a:schemeClr val="accent2"/>
    </p:to>
  </p:animClr>
</p:childTnLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

19.5.91 txEl (Text Element)

This element specifies a text element to animate.

[*Example*: Consider a shape containing text to be animated. The <txEl> should be used as follows:

```
<p:tgtEl>
  <p:spTgt spid="5">
    <p:txEl>
      <p:pRg st="1" end="1"/>
    </p:txEl>
  </p:spTgt>
</p:tgtEl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLTextTargetElement) is located in §A.3. *end note*]

19.5.92 val (Value)

The element specifies a value for a time animate.

[*Example*: Consider a shape with a fade in animation effect. The <val> element should be used as follows:

```
<p:anim calcmode="lin" valueType="num">
  <p:cBhvr additive="base"> … </p:cBhvr>
  <p:tavLst>
    <p:tav tm="0%">
      <p:val>
        <p:strVal val="0-#ppt_w/2"/>
      </p:val>
    </p:tav>
    <p:tav tm="100%">
      <p:val>
        <p:strVal val="#ppt_x"/>
      </p:val>
    </p:tav>
  </p:tavLst>
</p:anim>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLAnimVariant) is located in §A.3. *end note*]

19.5.93 video (Video)

This element specifies video information in an animation sequence. This element specifies that this node within the animation tree triggers the playback of a video file; the actual video file used is specified by the videoFile element (§20.1.3.6).

[*Example*: Consider a slide with an animated video content. The <video> element is used as follows:

```
<p:cSld>
  <p:spTree>
    <p:pic>
      <p:nvPicPr>
        <p:cNvPr id="4"/>
        …
        <p:nvPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <a:videoFile r:link="rId1" contentType="video/ogg"/>
          </p:nvPr>
        </p:nvPicPr>
        …
      </p:pic>
    </p:spTree>
  </p:cSld>
  …
  <p:childTnLst>
    <p:seq concurrent="1" nextAc="seek">
     …
    </p:seq>
    <p:video>
      <p:cMediaNode>
        …
        <p:tgtEl>
          <p:spTgt spid="4"/>
        </p:tgtEl>
      </p:cMediaNode>
    </p:video>
  </p:childTnLst>
```

The video element specifies the location of the video playback within the animation sequence; its child spTgt element specifies that the shape which contains the video to be played has a shape ID of 4. If we look at the shape with that ID value, its child videoFile element references an external video file of content type `video/ogg` located at the target of the relationship with ID `rId1`.*end example*]

Attributes	Description
fullScrn (Full Screen)	This attribute specifies if the video is displayed in full-screen. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TLMediaNodeVideo) is located in §A.3. *end note*]

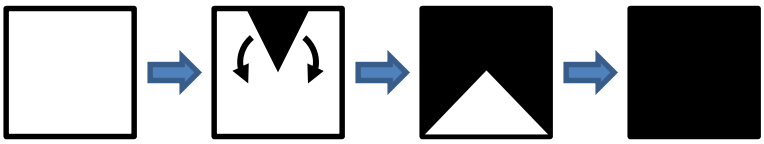
19.5.94 wedge (Wedge Slide Transition)

This element describes the wedge slide transition effect, which uses two radial edges that wipe from top to bottom in opposite directions until the new slide is fully shown. The rendering of this transition has been shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following case in which the “wedge” slide transition is applied to a slide, along with a set of attributes. The proper usage and a sample rendering are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:wedge/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

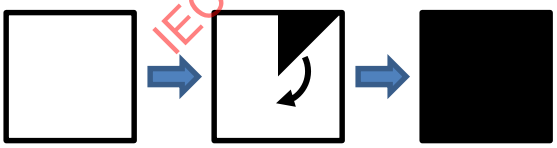
[*Note*: The W3C XML Schema definition of this element’s content model (CT_Empty) is located in §A.3. *end note*]

19.5.95 wheel (Wheel Slide Transition)

This element describes the wheel slide transition effect, which uses a set of radial edges and wipes them in the clockwise direction until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the “wheel” slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

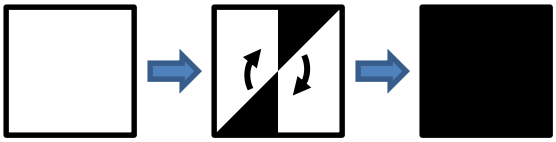
```
<p:transition>
  <p:wheel spokes="1"/>
</p:transition>
```



```
<p:transition>
  <p:wheel spokes="2"/>
</p:transition>
```

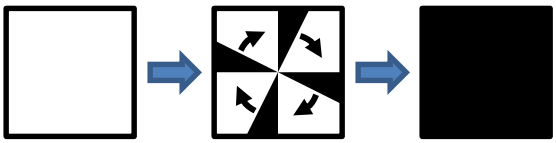
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

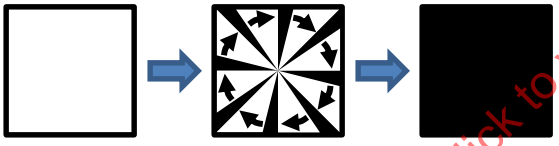


```
<p:transition>
  <p:wheel spokes="3"/>
</p:transition>
```

```
<p:transition>
  <p:wheel spokes="4"/>
</p:transition>
```



```
<p:transition>
  <p:wheel spokes="8"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
spokes (Spokes)	This attributes specifies the number of spokes ("pie pieces") in the wheel The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WheelTransition) is located in §A.3. *end note*]

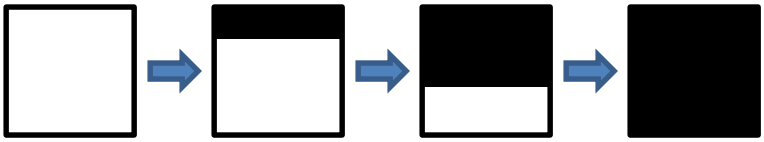
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.5.96 wipe (Wipe Slide Transition)

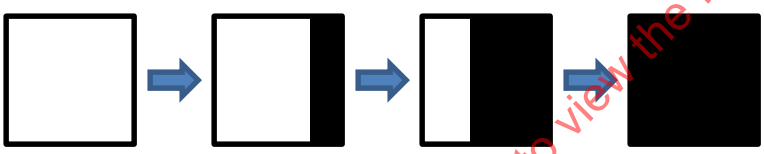
This element describes the wipe slide transition effect, which wipes the new slide over the previous slide from one edge of the screen to the opposite until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the “wipe” slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

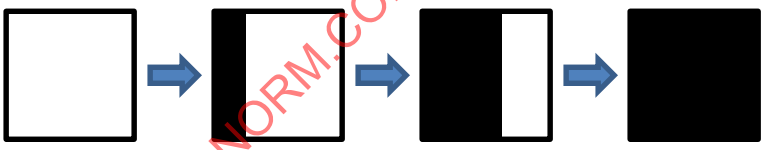
```
<p:transition>
  <p:wipe dir="d"/>
</p:transition>
```



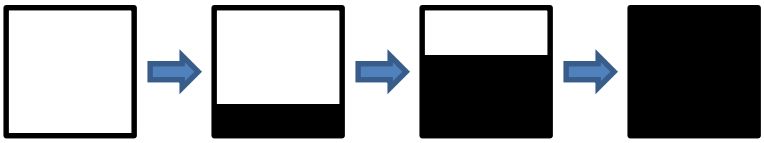
```
<p:transition>
  <p:wipe dir="l"/>
</p:transition>
```



```
<p:transition>
  <p:wipe dir="r"/>
</p:transition>
```



```
<p:transition>
  <p:wipe dir="u"/>
</p:transition>
```



end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

Attributes	Description
dir (Direction)	This attribute specifies the direction of the slide transition. The possible values for this attribute are defined by the ST_TransitionSideDirectionType simple type (§19.7.53).

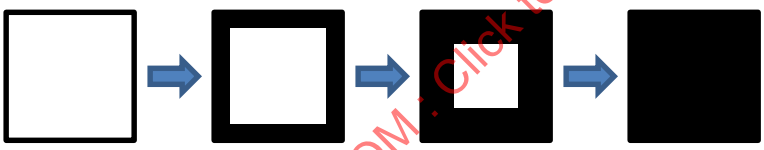
[*Note*: The W3C XML Schema definition of this element's content model (CT_SideDirectionTransition) is located in §A.3. *end note*]

19.5.97 zoom (Zoom Slide Transition)

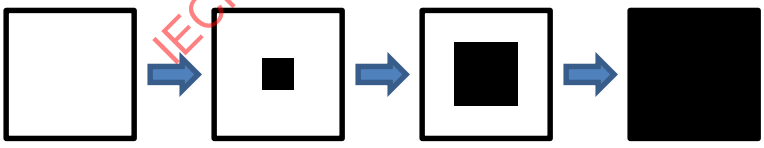
This element describes the zoom slide transition effect, which uses a box pattern centered on the slide that increases in size until the new slide is fully shown. The rendering of this transition depends upon the attributes specified which have been shown below.

[*Example*: Consider the following cases in which the "zoom" slide transition is applied to a slide, along with a set of attributes. The proper usage and sample renderings are shown below, with the XML fragments preceding the corresponding rendering:

```
<p:transition>
  <p:zoom dir="in"/>
</p:transition>
```



```
<p:transition>
  <p:zoom dir="out"/>
</p:transition>
```



end example]

[*Note*: Any rendering shown above is for example purposes only. Exact rendering of any transition is determined by the rendering application. As such, the same transition can have many variations depending on the implementation. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
dir (Direction)	This attribute specifies the direction of an "in/out" slide transition. The possible values for this attribute are defined by the ST_TransitionInOutDirectionType simple type (§19.7.52).

[*Note*: The W3C XML Schema definition of this element's content model (CT_InOutTransition) is located in §A.3. *end note*]

19.6 Slide Synchronization Data

It is often the case that slides are repurposed from existing presentations to be used in other presentations. In such cases, it is often beneficial for there to be an association, or a pairing, between the original slide and all copied instances of it. In the presence of such a pairing, applications can enable a variety of time-saving features, including the automatic updates of copied slides when the original slide changes. The Slide Synchronization Data part is designed to enable such application-defined functionality.

This information is stored in the Slide Synchronization Data part, which is referenced via an implicit relationship from the associated Slide part.

19.6.1 sldSyncPr (Slide Synchronization Properties)

This element specifies the information needed to associate the original slide with all copied instances of it.

Attributes	Description
clientInsertedTime (Client Slide Insertion date/time)	The date and time that the original slide was last updated in the current presentation. The date/time is stored in ISO 8601 format. [*Note*: This value can be used to inform the user of when the last synchronization was, as well as to determine when to next check for an updated version. *end note*] The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.
serverSldId (Server's Slide File ID)	A string that, when paired with the target of the Slide Synchronization Data part's external relationship, uniquely identifies the original slide. The possible values for this attribute are defined by the W3C XML Schema string datatype.
serverSldModifiedTime (Server's Slide File's modification	The date and time that the original slide was last modified in its location as defined by the target of the Slide Synchronization Data part's external relationship.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
date/time)	The date and time are stored in ISO 8601 format. The possible values for this attribute are defined by the W3C XML Schema dateTime datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SlideSyncProperties) is located in §A.3. *end note*]

19.7 Simple Types

This is the complete list of simple types dedicated to PresentationML.

19.7.1 ST_BookmarkIdSeed (Bookmark ID Seed)

This simple type specifies constraints for value of the Bookmark ID seed.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than 2147483648.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BookmarkIdSeed) is located in §A.3. *end note*]

19.7.2 ST_Direction (Direction)

This simple type defines a direction of either horizontal or vertical.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
horz (Horizontal)	Defines a horizontal direction.
vert (Vertical)	Defines a vertical direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Direction) is located in §A.3. *end note*]

19.7.3 ST_Index (Index)

This simple type defines the position of an object in an ordered list.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Index) is located in §A.3. *end note*]

19.7.4 ST_IterateType (Iterate Type)

This simple type specifies how the animation is applied over subelements of the target element.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
el (Element)	Iterate by element.
lt (Letter)	Iterate by Letter.
wd (Word)	Iterate by Word.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_IterateType) is located in §A.3. *end note*]

19.7.5 ST_Name (Name string)

This simple type specifies a name, such as for a comment author or custom show.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Name) is located in §A.3. *end note*]

19.7.6 ST_OleObjectFollowColorScheme (Embedded object to Follow Color Scheme)

This simple type determines if the Embedded object is re-colored to reflect changes to the color schemes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
full (Full)	Setting this enumeration causes the Embedded object to respond to all changes in the color scheme in the presentation.
none (None)	Setting this enumeration causes the Embedded object to not respond to changes in the color scheme in the presentation.
textAndBackground (Text and Background)	Setting this enumeration causes the Embedded object

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	to respond only to changes in the text and background colors of the color scheme in the presentation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OleObjectFollowColorScheme) is located in §A.3. *end note*]

19.7.7 ST_PhotoAlbumFrameShape (Photo Album Shape for Photo Mask)

This simple type specifies the values for photo frame types within a photo album presentation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
frameStyle1 (Rectangle Photo Frame)	
frameStyle2 (Rounded Rectangle Photo Frame)	
frameStyle3 (Simple White Photo Frame)	
frameStyle4 (Simple Black Photo Frame)	
frameStyle5 (Compound Black Photo Frame)	
frameStyle6 (Center Shadow Photo Frame)	
frameStyle7 (Soft Edge Photo Frame)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

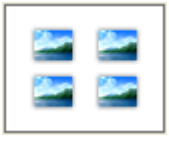
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PhotoAlbumFrameShape) is located in §A.3. *end note*]

19.7.8 ST_PhotoAlbumLayout (Photo Album Layout Definition)

This simple type specifies the values for photo layouts within a photo album presentation.

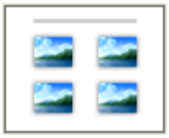
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1pic (1 Photo per Slide)	Specifies that photo album slides should have a single picture, centered horizontally and vertically, on the slide with no title.
1picTitle (1 Photo per Slide with Titles)	Specifies that photo album slides should have a single picture and a single title text box, centered horizontally and vertically, on the slide.
2pic (2 Photos per Slide)	Specifies that photo album slides should have two pictures of the same size, positioned side-by-side, centered horizontally and vertically, on the slide with no title.
2picTitle (2 Photos per Slide with Titles)	Specifies that photo album slides should have two pictures of the same size, positioned side-by-side, with a single title text box centered over them, collectively centered horizontally and vertically, on the slide.
4pic (4 Photos per Slide)	 Specifies that photo album slides should have four pictures of the same size, positioned in a two-by-two

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	matrix, centered horizontally and vertically, on the slide with no title.
4picTitle (4 Photos per Slide with Titles)	 Specifies that photo album slides should have four pictures of the same size, positioned in a two-by-two matrix, with a single title text box centered over the matrix, centered horizontally and vertically, on the slide.
fitToSlide (Fit Photos to Slide)	Specifies that photo album slides should have a single picture, stretched to fit the entire slide size, with no title.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PhotoAlbumLayout) is located in §A.3. *end note*]

19.7.9 ST_PlaceholderSize (Placeholder Size)

This simple type facilitates the storing of the size of the placeholder. This size is described relative to the body placeholder on the master.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
full (Full)	Specifies that the placeholder should take the full size of the body placeholder on the master.
half (Half)	Specifies that the placeholder should take the half size of the body placeholder on the master. Half size vertically or horizontally? Needs a picture.
quarter (Quarter)	Specifies that the placeholder should take a quarter of the size of the body placeholder on the master. Picture would be helpful

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PlaceholderSize) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

19.7.10 ST_PlaceholderType (Placeholder IDs)

This simple type facilitates the storing of the content type a placeholder should contain.

[*Note*: Some placeholder types are not allowed for all SlideBase types. *end note*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
body (Body)	Contains body text. Allowed for Slide, Slide Layout, Slide Master, Notes, Notes Master. Can be horizontal or vertical on Slide and Slide Layout.
chart (Chart)	Contains a chart or graph. Special type. Allowed for Slide and Slide Layout.
clipArt (Clip Art)	Contains a single clip art image. Special type. Allowed for Slide and Slide Layout.
ctrTitle (Centered Title)	Contains a title intended to be centered on the slide. Allowed for Slide and Slide Layout.
dgm (Diagram)	Contains a diagram. Special type. Allowed for Slide and Slide Layout.
dt (Date and Time)	Contains the date and time. Allowed for Slide, Slide Layout, Slide Master, Notes, Notes Master, Handout Master
ftr (Footer)	Contains text to be used as a footer in the document. Allowed for Slide, Slide Layout, Slide Master, Notes, Notes Master, Handout Master
hdr (Header)	Contains text to be used as a header for the document. Allowed for Notes, Notes Master, Handout Master .
media (Media)	Contains multimedia content such as audio or a movie clip. Special type. Allowed for Slide and Slide Layout.
obj (Object)	Contains any content type. Special type. Allowed for Slide and Slide Layout.
pic (Picture)	Contains a picture. Special type. Allowed for Slide and Slide Layout.
sldImg (Slide Image)	Contains an image of the slide. Allowed for Notes and Notes Master.
sldNum (Slide Number)	Contains the number of a slide. Allowed for Slide, Slide Layout, Slide Master, Notes, Notes Master, Handout Master
subTitle (Subtitle)	Contains a subtitle. Allowed for Slide and Slide Layout.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
tbl (Table)	Contains a table. Special type. Allowed for Slide and Slide Layout.
title (Title)	Contains a slide title. Allowed for Slide, Slide Layout and Slide Master. Can be horizontal or vertical on Slide and Slide Layout.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PlaceholderType) is located in §A.3. *end note*]

19.7.11 ST_PrintColorMode (Print Color Mode)

This simple type specifies the color mode that should be used when printing a presentation document.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bw (Black and White Mode)	Print should be in Black and White only
clr (Color Mode)	Print should be in Full Color
gray (Grayscale Mode)	Print should be in Grayscale only

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PrintColorMode) is located in §A.3. *end note*]

19.7.12 ST_PrintWhat (Default print output)

This simple type specifies the default print layout that should be used when printing

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
handouts1 (1 Slide / Handout Page)	1 Slide and Handout Page layout should be used.
handouts2 (2 Slides / Handout Page)	2 Slides and Handout Page layout should be used.
handouts3 (3 Slides / Handout Page)	3 Slides and Handout Page layout should be used.
handouts4 (4 Slides / Handout Page)	4 Slides and Handout Page layout should be used.
handouts6 (6 Slides / Handout Page)	6 Slides and Handout Page layout should be used.
handouts9 (9 Slides / Handout Page)	9 Slides and Handout Page layout should be used.
notes (Notes)	Notes layout should be used.
outline (Outline)	Outline layout should be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
slides (Slides)	Slides layout should be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PrintWhat) is located in §A.3. *end note*]

19.7.13 ST_SlideId (Slide Identifier)

This simple type specifies the allowed numbering for the slide identifier.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a maximum value of less than 2147483648.
- This simple type has a minimum value of greater than or equal to 256.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideId) is located in §A.3. *end note*]

19.7.14 ST_SlideLayoutId (Slide Layout ID)

This simple type sets the bounds for the slide layout id value. This layout id is used to identify the different slide layout designs.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 2147483648.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideLayoutId) is located in §A.3. *end note*]

19.7.15 ST_SlideLayoutType (Slide Layout Type)

This simple type defines an arrangement of content on a slide. Each layout type is not tied to an exact positioning of placeholders, but rather provides a higher-level description of the content type and positioning of placeholders. This information can be used by the application to aid in mapping between different layouts. The application can choose which, if any, of these layouts to make available through its user interface.

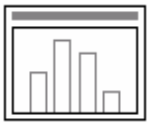
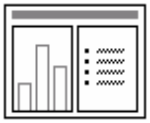
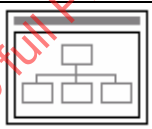
Each layout contains zero or more placeholders, each with a specific content type. An "object" placeholder can contain any kind of data. Media placeholders are intended to hold video or audio clips. The enumeration value descriptions include illustrations of sample layouts for each value of the simple type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

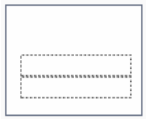
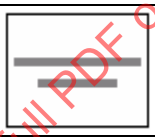
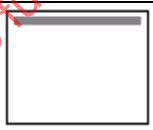
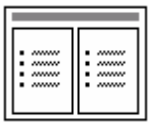
 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
blank (Slide Layout Type Enumeration (Blank))	Blank
chart (Chart)	 Title and chart
chartAndTx (Slide Layout Type Enumeration (Chart and Text))	 Title, chart on left and text on right
clipArtAndTx (Clip Art and Text)	Title, clipart on left, text on right
clipArtAndVertTx (Clip Art and Vertical Text)	Title, clip art on left, vertical text on right
cust (Slide Layout Type Enumeration (Custom))	Custom layout defined by user
dgm (Slide Layout Type Enumeration (Diagram))	 Title and diagram
fourObj (Four Objects)	Title and four objects
mediaAndTx (Slide Layout Type Enumeration (Media and Text))	Title, media on left, text on right
obj (Title and Object)	Title and object
objAndTwoObj (Object and Two Object)	Title, one object on left, two objects on right
objAndTx (Slide Layout Type Enumeration (Object and Text))	Title, object on left, text on right
objOnly (Object)	Object only

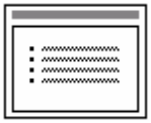
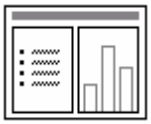
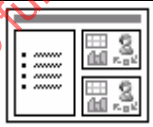
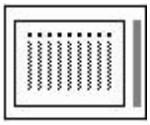
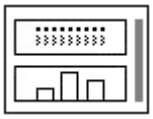
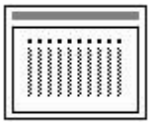
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
objOverTx (Slide Layout Type Enumeration (Object over Text))	Title, object on top, text on bottom
objTx (Title, Object, and Caption)	Title, object and caption text
picTx (Picture and Caption)	Title, picture, and caption text
secHead (Section Header)	 Section header title and subtitle text
tbl (Slide Layout Type Enumeration (Table))	Title and table
title (Slide Layout Type Enumeration (Title))	 Title layout with centered title and subtitle placeholders
titleOnly (Slide Layout Type Enumeration (Title Only))	 Title only
twoColTx (Slide Layout Type Enumeration (Two Column Text))	 Title, text on left, text on right
twoObj (Two Objects)	Title, object on left, object on right
twoObjAndObj (Two Objects and Object)	Title, two objects on left, one object on right
twoObjAndTx (Two Objects and Text)	Title, two objects on left, text on right
twoObjOverTx (Two Objects over Text)	Title, two objects on top, text on bottom

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
twoTxTwoObj (Two Text and Two Objects)	Title, two objects each with text
tx (Slide Layout Type Enumeration (Text))	 Title and text
txAndChart (Slide Layout Type Enumeration (Text and Chart))	 Title, text on left and chart on right
txAndClipArt (Text and Clip Art)	Title, text on left, clip art on right
txAndMedia (Slide Layout Type Enumeration (Text and Media))	Title, text on left, media on right
txAndObj (Slide Layout Type Enumeration (Text and Object))	Title, text on left, object on right
txAndTwoObj (Text and Two Objects)	 Title, text on left, two objects on right
txOverObj (Slide Layout Type Enumeration (Text over Object))	Title, text on top, object on bottom
vertTitleAndTx (Vertical Title and Text)	 Vertical title on right, vertical text on left
vertTitleAndTxOverChart (Vertical Title and Text Over Chart)	 Vertical title on right, vertical text on top, chart on bottom
vertTx (Vertical Text)	 Title and vertical text body

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideLayoutType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.7.16 ST_SlideMasterId (Slide Master ID)

This simple type specifies the allowed numbering for the slide master identifier.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 2147483648.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideMasterId) is located in §A.3. *end note*]

19.7.17 ST_SlideSizeCoordinate (Slide Size Coordinate)

This simple type specifies the slide size coordinate in EMUs (English Metric Units).

This simple type's contents are a restriction of the ST_PositiveCoordinate32 datatype (§20.1.10.43).

This simple type also specifies the following restrictions:

- This simple type has a maximum value of less than or equal to 51206400.
- This simple type has a minimum value of greater than or equal to 914400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideSizeCoordinate) is located in §A.3. *end note*]

19.7.18 ST_SlideSizeType (Slide Size Type)

This simple type specifies the kind of slide size that the slide should be optimized for.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
35mm (35mm Film)	Slide size should be optimized for 35mm film output
A3 (A3)	Slide size should be optimized for A3 output
A4 (A4)	Slide size should be optimized for A4 output
B4ISO (B4ISO)	Slide size should be optimized for B4ISO output
B4JIS (B4JIS)	Slide size should be optimized for B4JIS output
B5ISO (B5ISO)	Slide size should be optimized for B5ISO output
B5JIS (B5JIS)	Slide size should be optimized for B5JIS output
banner (Banner)	Slide size should be optimized for banner output
custom (Custom)	Slide size should be optimized for custom output
hagakiCard (Hagaki Card)	Slide size should be optimized for hagaki card output

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
ledger (Ledger)	Slide size should be optimized for ledger output
letter (Letter)	Slide size should be optimized for letter output
overhead (Overhead)	Slide size should be optimized for overhead output
screen16x10 (Screen 16x10)	Slide size should be optimized for 16x10 screen output
screen16x9 (Screen 16x9)	Slide size should be optimized for 16x9 screen output
screen4x3 (Screen 4x3)	Slide size should be optimized for 4x3 screen output

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SlideSizeType) is located in §A.3. *end note*]

19.7.19 ST_SplitterBarState (Splitter Bar State)

This simple type specifies the state that the splitter bar should be shown in. The splitter bar separates a primary and secondary region within a viewing area.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
maximized (Max)	The primary region occupies the greatest amount of the viewing area allowed by the application.
minimized (Min)	The primary region occupies the least amount of the viewing area allowed by the application.
restored (Restored)	The primary region has a specific intermediate size.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SplitterBarState) is located in §A.3. *end note*]

19.7.20 ST_TLAnimateBehaviorCalcMode (Time List Animate Behavior Calculate Mode)

This simple type specifies how the animation flows from point to point.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
discrete (Calc Mode Enum (Discrete))	Descrete
fmla (Calc Mode Enum (Formula))	Formula
lin (Calc Mode Enum (Linear))	Linear

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateBehaviorCalcMode) is located in §A.3. *end note*]

19.7.21 ST_TLAnimateBehaviorValueType (Time List Animate Behavior Value Types)

This simple type specifies the type of property value.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clr (Value Type Enum (Color))	Color
num (Value Type Enum (Number))	Number
str (Value Type Enum (String))	String

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateBehaviorValueType) is located in §A.3. *end note*]

19.7.22 ST_TLAnimateColorDirection (Time List Animate Color Direction)

This simple type specifies the direction in which to interpolate the animation (clockwise or counterclockwise).

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ccw (Counter-Clockwise)	Counter-Clockwise
cw (Direction Enum (Clockwise))	Clockwise

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateColorDirection) is located in §A.3. *end note*]

19.7.23 ST_TLAnimateColorSpace (Time List Animate Color Space)

This simple type specifies the color space of the animation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hsl (Color Space Enum (HSL))	Hue, Saturation, Luminance
rgb (Color Space Enum (RGB))	Red, Green, Blue

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateColorSpace) is located in §A.3. *end note*]

19.7.24 ST_TLAnimateEffectTransition (Time List Animate Effect Transition)

This simple type specifies whether the effect is a transition in, transition out, or neither.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
in (Transition Enum (In))	In
none (Transition Enum (None))	None
out (Transition Enum (Out))	Out

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateEffectTransition) is located in §A.3. *end note*]

19.7.25 ST_TLAnimateMotionBehaviorOrigin (Time List Animate Motion Behavior Origin)

This simple type specifies what the origin of the motion path is relative to.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
layout (Origin Enum (Layout))	Layout
parent (Origin Enum (Parent))	Parent

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateMotionBehaviorOrigin) is located in §A.3. *end note*]

19.7.26 ST_TLAnimateMotionPathEditMode (Time List Animate Motion Path Edit Mode)

This simple type specifies how the motion path moves when the target element is moved.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fixed (Path Edit Mode Enum (Fixed))	Fixed
relative (Path Edit Mode Enum (Relative))	Relative

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLAnimateMotionPathEditMode) is located in §A.3. *end note*]

19.7.27 ST_TLBehaviorAccumulateType (Behavior Accumulate Type)

This simple type makes a repeating animation build with each iteration when set to "always."

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
always (Accumulate Enum (Always))	Always
none (Accumulate Enum (None))	None

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLBehaviorAccumulateType) is located in §A.3. *end note*]

19.7.28 ST_TLBehaviorAdditiveType (Behavior Additive Type)

This simple type specifies how to apply the animation values to the original value for the property.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
base (Additive Enum (Base))	Base
mult (Additive Enum (Multiply))	Multiply
none (None)	None
repl (Additive Enum (Replace))	Replace
sum (Additive Enum (Sum))	Sum

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLBehaviorAdditiveType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.7.29 ST_TLBehaviorOverrideType (Behavior Override Type)

This simple type specifies how a behavior should override values of the attribute being animated on the target element. The "childStyle" clears the attributes on the children contained inside the target element.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
childStyle (Override Enum (Child Style))	Child Style
normal (Override Enum (Normal))	Normal

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLBehaviorOverrideType) is located in §A.3. *end note*]

19.7.30 ST_TLBehaviorTransformType (Behavior Transform Type)

This simple type specifies how the behavior animates the target element.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
img (Image)	Image transform
pt (Point)	Point transform

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLBehaviorTransformType) is located in §A.3. *end note*]

19.7.31 ST_TLChartSubelementType (Chart Subelement Type)

This simple type defines an animation target element that is represented by a subelement of a chart.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
category (Category Axis)	Category
gridLegend (Grid Legend)	Background Element (Grid and Legend)
ptInCategory (Single Point in Category)	Category Element
ptInSeries (Single Point in Data Series)	Series Element
series (Data Series)	Series

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLChartSubelementType) is located in §A.3. *end note*]

19.7.32 ST_TLCommandType (Command Type)

This simple type specifies a command type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
call (Command Type Enum (Call))	Call
evt (Command Type Enum (Event))	Event
verb (Command Type Enum (Verb))	Verb

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLCommandType) is located in §A.3. *end note*]

19.7.33 ST_TLDiagramBuildType (Diagram Build Types)

This simple type specifies the different diagram build types.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allAtOnce (Diagram Build Type Enum (All At Once))	All At Once
breadthByLvl (Diagram Build Type Enum (Breadth By Level))	Breadth By Level
breadthByNode (Diagram Build Type Enum (Breadth By Node))	Breadth By Node
ccw (Diagram Build Type Enum (Counter-Clockwise))	Counter-Clockwise
ccwIn (Diagram Build Type Enum (Counter-Clockwise-In))	Counter-Clockwise-In
ccwOut (Diagram Build Type Enum (Counter-Clockwise-Out))	Counter-Clockwise-Out
cust (Diagram Build Type Enum (Custom))	Custom
cw (Diagram Build Type Enum (Clockwise))	Clockwise
cwIn (Diagram Build Type Enum (Clockwise-In))	Clockwise-In
cwOut (Diagram Build Type Enum (Clockwise-Out))	Clockwise-Out

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
depthByBranch (Diagram Build Type Enum (Depth By Branch))	Depth By Branch
depthByNode (Diagram Build Type Enum (Depth By Node))	Depth By Node
down (Diagram Build Type Enum (Down))	Down
inByRing (Diagram Build Type Enum (In-By-Ring))	In-By-Ring
outByRing (Diagram Build Type Enum (Out-By-Ring))	Out-By-Ring
up (Diagram Build Type Enum (Up))	Up
whole (Diagram Build Type Enum (Whole))	Whole

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLDiagramBuildType) is located in §A.3. *end note*]

19.7.34 ST_TLNextActionType (Next Action Type)

This simple type specifies what to do when going forward in a sequence. When the value is "seek," it seeks the current child element to its natural end time before advancing to the next element.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (Next Action Type Enum (None))	None
seek (Next Action Type Enum (Seek))	Seek

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLNextActionType) is located in §A.3. *end note*]

19.7.35 ST_TLOleChartBuildType (Embedded Chart Build Type)

This simple type describes how to build an embedded Chart.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allAtOnce (Chart Build Type Enum (All At Once))	All At Once
category (Chart Build Type Enum (Category))	By Category
categoryEl (Chart Build Type Enum (Category Element))	By Category Element

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
series (Chart Build Type Enum (Series))	By Series
seriesEl (Chart Build Type Enum (Series Element))	By Series Element

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLOleChartBuildType) is located in §A.3. *end note*]

19.7.36 ST_TLParaBuildType (Paragraph Build Type)

This simple type describes how to build a paragraph.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allAtOnce (All At Once)	Specifies to animate all paragraphs at once.
cust (Custom)	Specifies the build has custom user settings.
p (Paragraph)	Specifies to animate paragraphs grouped by bullet level.
whole (Whole)	Specifies to animate the entire body of text as one block.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLParaBuildType) is located in §A.3. *end note*]

19.7.37 ST_TLPreviousActionType (Previous Action Type)

This simple type specifies what to do when going backwards in a sequence. When the value is "skipTimed," the sequence continues to go backwards until it reaches a sequence element that was defined to being only on a "next" event.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (Previous Action Type Enum (None))	None
skipTimed (Previous Action Type Enum (Skip Timed))	Skip Timed

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLPreviousActionType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.7.38 ST_TLTime (Time)

This simple type specifies time after which to automatically advance the build to the next step. An amount of time, in milliseconds.

This simple type is a union of the following types:

- The ST_TLTimeIndefinite simple type (§19.7.40).
- The W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTime) is located in §A.3. *end note*]

19.7.39 ST_TLTimeAnimateValueTime (Animation Time)

This simple type specifies a percentage within the time span of the element. A value of `indefinite` means the attribute should be ignored.

This simple type is a union of the following types:

- The ST_PositiveFixedPercentage simple type (§20.1.10.45).
- The ST_TLTimeIndefinite simple type (§19.7.40).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeAnimateValueTime) is located in §A.3. *end note*]

19.7.40 ST_TLTimeIndefinite (Indefinite Time Declaration)

This simple type specifies a value that designates an "indefinite" amount time -- typically means this property is subordinate to other, defined properties.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
indefinite (Indefinite Type Enum)	Specifies Indefinite Time

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeIndefinite) is located in §A.3. *end note*]

19.7.41 ST_TLTimeNodeFillType (Time Node Fill Type)

This simple type specifies what modifications the effect leaves on the target element's properties when the effect ends.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
freeze (Freeze)	Freeze
hold (TimeNode Fill Type Enum (Hold))	Hold
remove (Remove)	Remove
transition (Transition)	Transition

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeFillType) is located in §A.3. *end note*]

19.7.42 ST_TLTimeNodeID (Time Node ID)

This simple type represents a node or event on the timeline by its identifier.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeID) is located in §A.3. *end note*]

19.7.43 ST_TLTimeNodeMasterRelation (Time Node Master Relation)

This simple type specifies how the time node plays back relative to its master time node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lastClick (TimeNode Master Relation Enum (Last Click))	Last Click
nextClick (TimeNode Master Relation Enum (Next Click))	Next Click
sameClick (TimeNode Master Relation Enum (Same Click))	Same Click

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeMasterRelation) is located in §A.3. *end note*]

19.7.44 ST_TLTimeNodePresetClassType (Time Node Preset Class Type)

This simple type specifies the class of effect in which this effect belongs.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
emph (Preset Type Enum (Emphasis))	Emphasis Preset
entr (Preset Type Enum (Entrance))	Entrance Preset
exit (Exit)	Exit Preset
mediacall (Preset Type Enum (Media Call))	Media Call Preset
path (Preset Type Enum (Path))	Path Preset
verb (Preset Type Enum (Verb))	Verb Preset

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodePresetClassType) is located in §A.3. *end note*]

19.7.45 ST_TLTimeNodeRestartType (Time Node Restart Type)

This simple type determines whether an effect can play more than once.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
always (Restart Enum (Always))	Always restart node
never (Restart Enum (Never))	Never restart node
whenNotActive (Restart Enum (When Not Active))	Restart when node is not active

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeRestartType) is located in §A.3. *end note*]

19.7.46 ST_TLTimeNodeSyncType (Time Node Sync Type)

This simple type specifies how the time node synchronizes to its group.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
canSlip (TimeNode Sync Enum (Can Slip))	Can Slip
locked (TimeNode Sync Enum (Locked))	Locked

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeSyncType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.7.47 ST_TLTimeNodeType (Time Node Type)

This simple type specifies time node types.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
afterEffect (Node Type Enum (After Effect))	After Effect
afterGroup (Node Type Enum (After Group))	After Group
clickEffect (Node Type Enum (Click Effect))	Click Effect
clickPar (Node Type Enum (Click Paragraph))	Click Paragraph
interactiveSeq (Node Type Enum (Interactive Sequence))	Interactive Sequence
mainSeq (Node Type Enum (Main Sequence))	Main Sequence
tmRoot (Node Type Enum (Timing Root))	Timing Root
withEffect (Node Type Enum (With Effect))	With Effect
withGroup (Node Type Enum (With Group))	With Group

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTimeNodeType) is located in §A.3. *end note*]

19.7.48 ST_TLTriggerEvent (Trigger Event)

This simple type specifies a particular event that causes the time condition to be true.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
begin (Trigger Event Enum (Begin))	Fire trigger at the beginning
end (Trigger Event Enum (End))	Fire trigger at the end
onBegin (Trigger Event Enum (On Begin))	Fire trigger at the beginning
onClick (Trigger Event Enum (On Click))	Fire trigger on a mouse click
onDblClick (Trigger Event Enum (On Double Click))	Fire trigger on double-mouse click
onEnd (Trigger Event Enum (On End))	Fire trigger at the end
onMouseOut (Trigger Event Enum (On Mouse Out))	Fire trigger on mouse out
onMouseOver (Trigger Event Enum (On Mouse Over))	Fire trigger on mouse over
onNext (Trigger Event Enum (On Next))	Fire trigger on next node

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
onPrev (Trigger Event Enum (On Previous))	Fire trigger on previous node
onStopAudio (Trigger Event Enum (On Stop Audio))	Fire trigger on stop audio

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTriggerEvent) is located in §A.3. *end note*]

19.7.49 ST_TLTriggerRuntimeNode (Trigger RunTime Node)

This simple type specifies the child time node that triggers a time condition. References a child TimeNode or all child nodes. Order is based on the child's end time.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (Trigger RunTime Node Enum (All))	All
first (Trigger RunTime Node (First))	First
last (Trigger RunTime Node (Last))	Last

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TLTriggerRuntimeNode) is located in §A.3. *end note*]

19.7.50 ST_TransitionCornerDirectionType (Transition Corner Direction Type)

This simple type specifies diagonal directions for slide transitions.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ld (Transition Corner Direction Enum (Left-Down))	Specifies the slide transition direction of left-down
lu (Transition Corner Direction Enum (Left-Up))	Specifies the slide transition direction of left-up
rd (Transition Corner Direction Enum (Right-Down))	Specifies the slide transition direction of right-down
ru (Transition Corner Direction Enum (Right-Up))	Specifies the slide transition direction of right-up

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TransitionCornerDirectionType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

19.7.51 ST_TransitionEightDirectionType (Transition Eight Direction)

This simple type specifies the direction of an animation.

This simple type is a union of the following types:

- The ST_TransitionCornerDirectionType simple type (§19.7.50).
- The ST_TransitionSideDirectionType simple type (§19.7.53).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TransitionEightDirectionType) is located in §A.3. *end note*]

19.7.52 ST_TransitionInOutDirectionType (Transition In/Out Direction Type)

This simple type specifies if a slide transition should go in or out.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
in (Transition In/Out Direction Enum (In))	Specifies the slide transition should go in
out (Transition In/Out Direction Enum (Out))	Specifies the slide transition should go out

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TransitionInOutDirectionType) is located in §A.3. *end note*]

19.7.53 ST_TransitionSideDirectionType (Transition Side Direction Type)

This simple type defines a set of slide transition directions.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
d (Transition Side Direction Enum (Down))	Specifies that the transition direction is down
l (Transition Side Direction Enum (Left))	Specifies that the transition direction is left
r (Transition Side Direction (Right))	Specifies that the transition direction is right
u (Transition Side Direction Enum (Up))	Specifies that the transition direction is up

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TransitionSideDirectionType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

19.7.54 ST_TransitionSpeed (Transition Speed)

This simple type defines the allowed transition speeds for transitioning from the current slide to the next.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fast (Fast)	Fast slide transition.
med (Medium)	Medium slide transition.
slow (low)	Slow slide transition.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TransitionSpeed) is located in §A.3. *end note*]

19.7.55 ST_ViewType (List of View Types)

This simple type defines the kinds of view available to an application when rendering a PresentationML document. Those view kinds are, as follows: *handout view*, *normal slide view*, *notes master view*, *notes view*, *outline view*, *slide master view*, *slide sorter view*, and *slide thumbnail view*. [*Note*: Although this Standard is for a file format, occasionally, guidance is given regarding intent in dealing with things outside that file format, such as the rendering of documents to a screen or printer. *end note*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
handoutView (Handout Master View)	Specifies that a given PresentationML document should be rendered using some sort of template that is intended to facilitate the changing of the design and layout of printed handouts.
notesMasterView (Notes Master View)	Specifies that a given PresentationML document should be rendered using some sort of template that is intended to facilitate the changing of the design and layout of notes slides.
notesView (Notes View)	Specifies that a given PresentationML document should be rendered using some sort of template that is intended to facilitate the viewing or editing of notes.
outlineView (Outline View)	Specifies that a given PresentationML document should be rendered in a view that is intended to facilitate the viewing of slides in some outline form.
sldMasterView (Slide Master View)	Specifies that a given PresentationML document should be rendered using some sort of template that is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	intended to facilitate the changing of the design and layout of master slides.
sldSorterView (Slide Sorter View)	Specifies that a given PresentationML document should be rendered in a view that is intended to facilitate the rearrangement of slides.
sldThumbnailView (Slide Thumbnail View)	Specifies that a given PresentationML document should be rendered in a view that shows slides in some thumbnail form.
sldView (Normal Slide View)	Specifies that a given PresentationML document should be rendered in a view that allows slides to be viewed or edited.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ViewType) is located in §A.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20. DrawingML - Framework Reference Material

[*Note*: For further information on the mapping of elements and attributes to OPC parts, see the Bibliography entry, “Information on elements, attributes, and OPC parts in ISO/IEC 29500 (OOXML)”. *end note*]

The subordinate subclauses specify the semantics for the XML markup comprising DrawingML content, which can be used within the contents of WordprocessingML, SpreadsheetML, or PresentationML documents.

This portion of DrawingML defines its core pieces.

20.1 DrawingML - Main

The DrawingML Main namespace defines all of the base constructs for all kinds of DrawingML objects (charts, diagrams, shapes, pictures, and so on). These constructs and primitives are defined below.

20.1.1 Table of Contents

This subclause is informative.

20.1.2 Basics 2727
20.1.2.1 EMU Unit of Measurement 2727
20.1.2.2 Core Drawing Object Information 2727
20.1.2.2.1 bldChart (Build Chart) 2727
20.1.2.2.2 bldDgm (Build Diagram) 2728
20.1.2.2.3 chart (Chart to Animate) 2729
20.1.2.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties) 2729
20.1.2.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties) 2730
20.1.2.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties) 2730
20.1.2.2.7 cNvPicPr (Non-Visual Picture Drawing Properties) 2730
20.1.2.2.8 cNvPr (Non-Visual Drawing Properties) 2731
20.1.2.2.9 cNvSpPr (Non-Visual Shape Drawing Properties) 2733
20.1.2.2.10 cxnSp (Connection Shape) 2734
20.1.2.2.11 cxnSpLocks (Connection Shape Locks) 2735
20.1.2.2.12 dgm (Diagram to Animate) 2737
20.1.2.2.13 endCxn (Connection End) 2737
20.1.2.2.14 ext (Extension) 2737
20.1.2.2.15 extLst (Extension List) 2738
20.1.2.2.16 graphic (Graphic Object) 2738
20.1.2.2.17 graphicData (Graphic Object Data) 2738
20.1.2.2.18 graphicFrame (Graphic Frame) 2739
20.1.2.2.19 graphicFrameLocks (Graphic Frame Locks) 2739
20.1.2.2.20 grpSp (Group shape) 2740
20.1.2.2.21 grpSpLocks (Group Shape Locks) 2741

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.2.2.22 grpSpPr (Visual Group Shape Properties) 2742
20.1.2.2.23 hlinkHover (Hyperlink for Hover) 2743
20.1.2.2.24 ln (Outline) 2744
20.1.2.2.25 nvCxnSpPr (Non-Visual Properties for a Connection Shape) 2745
20.1.2.2.26 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame) 2745
20.1.2.2.27 nvGrpSpPr (Non-Visual Properties for a Group Shape) 2745
20.1.2.2.28 nvPicPr (Non-Visual Properties for a Picture) 2746
20.1.2.2.29 nvSpPr (Non-Visual Properties for a Shape) 2746
20.1.2.2.30 pic (Picture) 2746
20.1.2.2.31 picLocks (Picture Locks) 2747
20.1.2.2.32 snd (Hyperlink Sound) 2749
20.1.2.2.33 sp (Shape) 2749
20.1.2.2.34 spLocks (Shape Locks) 2750
20.1.2.2.35 spPr (Shape Properties) 2751
20.1.2.2.36 stCxn (Connection Start) 2752
20.1.2.2.37 style (Shape Style) 2752
20.1.2.2.38 sx (Horizontal Ratio) 2752
20.1.2.2.39 sy (Vertical Ratio) 2753
20.1.2.2.40 txBody (Shape Text Body) 2753
20.1.2.2.41 txSp (Text Shape) 2753
20.1.2.2.42 useSpRect (Use Shape Text Rectangle) 2754
20.1.2.2.43 cpLocks (Content Part Locks) 2754
20.1.2.3 Colors 2755
20.1.2.3.1 alpha (Alpha) 2755
20.1.2.3.2 alphaMod (Alpha Modulation) 2756
20.1.2.3.3 alphaOff (Alpha Offset) 2757
20.1.2.3.4 blue (Blue) 2757
20.1.2.3.5 blueMod (Blue Modulation) 2758
20.1.2.3.6 blueOff (Blue Offset) 2758
20.1.2.3.7 comp (Complement) 2759
20.1.2.3.8 gamma (Gamma) 2760
20.1.2.3.9 gray (Gray) 2760
20.1.2.3.10 green (Green) 2760
20.1.2.3.11 greenMod (Green Modulation) 2760
20.1.2.3.12 greenOff (Green Offset) 2761
20.1.2.3.13 hslClr (Hue, Saturation, Luminance Color Model) 2762
20.1.2.3.14 hue (Hue) 2762
20.1.2.3.15 hueMod (Hue Modulate) 2763
20.1.2.3.16 hueOff (Hue Offset) 2764
20.1.2.3.17 inv (Inverse) 2764
20.1.2.3.18 invGamma (Inverse Gamma) 2765
20.1.2.3.19 lum (Luminance) 2765
20.1.2.3.20 lumMod (Luminance Modulation) 2766
20.1.2.3.21 lumOff (Luminance Offset) 2766
20.1.2.3.22 prstClr (Preset Color) 2767
20.1.2.3.23 red (Red) 2767
20.1.2.3.24 redMod (Red Modulation) 2768
20.1.2.3.25 redOff (Red Offset) 2768

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.2.3.26 sat (Saturation) 2769
20.1.2.3.27 satMod (Saturation Modulation) 2770
20.1.2.3.28 satOff (Saturation Offset) 2771
20.1.2.3.29 schemeClr (Scheme Color) 2771
20.1.2.3.30 scrgbClr (RGB Color Model - Percentage Variant) 2772
20.1.2.3.31 shade (Shade) 2772
20.1.2.3.32 srgbClr (RGB Color Model - Hex Variant) 2773
20.1.2.3.33 sysClr (System Color) 2774
20.1.2.3.34 tint (Tint) 2774

20.1.3 Audio and Video 2775

20.1.3.1 audioCd (Audio from CD) 2775
20.1.3.2 audioFile (Audio from File) 2776
20.1.3.3 end (Audio End Time) 2777
20.1.3.4 quickTimeFile (QuickTime from File) 2778
20.1.3.5 st (Audio Start Time) 2779
20.1.3.6 videoFile (Video from File) 2779
20.1.3.7 wavAudioFile (Audio from WAV File) 2780

20.1.4 Styles 2782

20.1.4.1 Styles 2782
20.1.4.1.1 accent1 (Accent 1) 2782
20.1.4.1.2 accent2 (Accent 2) 2782
20.1.4.1.3 accent3 (Accent 3) 2783
20.1.4.1.4 accent4 (Accent 4) 2784
20.1.4.1.5 accent5 (Accent 5) 2784
20.1.4.1.6 accent6 (Accent 6) 2785
20.1.4.1.7 bgFillStyleLst (Background Fill Style List) 2785
20.1.4.1.8 custClr (Custom color) 2786
20.1.4.1.9 dk1 (Dark 1) 2786
20.1.4.1.10 dk2 (Dark 2) 2787
20.1.4.1.11 effectStyle (Effect Style) 2788
20.1.4.1.12 effectStyleLst (Effect Style List) 2788
20.1.4.1.13 fillStyleLst (Fill Style List) 2789
20.1.4.1.14 fmtScheme (Format Scheme) 2790
20.1.4.1.15 folHlink (Followed Hyperlink) 2791
20.1.4.1.16 font (Font) 2791
20.1.4.1.17 fontRef (Font Reference) 2792
20.1.4.1.18 fontScheme (Font Scheme) 2792
20.1.4.1.19 hlink (Hyperlink) 2793
20.1.4.1.20 lnDef (Line Default) 2794
20.1.4.1.21 lnStyleLst (Line Style List) 2794
20.1.4.1.22 lt1 (Light 1) 2795
20.1.4.1.23 lt2 (Light 2) 2796
20.1.4.1.24 majorFont (Major Font) 2797
20.1.4.1.25 minorFont (Minor fonts) 2797
20.1.4.1.26 scene3d (3D Scene Properties) 2798
20.1.4.1.27 spDef (Shape Default) 2798
20.1.4.1.28 txDef (Text Default) 2799

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.4.2 Table Styles 2800
20.1.4.2.1 band1H (Band 1 Horizontal) 2800
20.1.4.2.2 band1V (Band 1 Vertical) 2801
20.1.4.2.3 band2H (Band 2 Horizontal) 2802
20.1.4.2.4 band2V (Band 2 Vertical) 2803
20.1.4.2.5 bevel (Bevel) 2804
20.1.4.2.6 bottom (Bottom Border) 2805
20.1.4.2.7 effect (Effect) 2805
20.1.4.2.8 effectRef (Effect Reference) 2805
20.1.4.2.9 fill (Fill) 2806
20.1.4.2.10 fillRef (Fill Reference) 2807
20.1.4.2.11 firstCol (First Column) 2807
20.1.4.2.12 firstRow (First Row) 2808
20.1.4.2.13 font (Font) 2809
20.1.4.2.14 insideH (Inside Horizontal Border) 2809
20.1.4.2.15 insideV (Inside Vertical Border) 2810
20.1.4.2.16 lastCol (Last Column) 2810
20.1.4.2.17 lastRow (Last Row) 2811
20.1.4.2.18 left (Left Border) 2812
20.1.4.2.19 lnRef (Line Reference) 2813
20.1.4.2.20 neCell (Northeast Cell) 2813
20.1.4.2.21 nwCell (Northwest Cell) 2814
20.1.4.2.22 right (Right Border) 2814
20.1.4.2.23 seCell (Southeast Cell) 2815
20.1.4.2.24 swCell (Southwest Cell) 2815
20.1.4.2.25 tblBg (Table Background) 2816
20.1.4.2.26 tblStyle (Table Style) 2817
20.1.4.2.27 tblStyleLst (Table Style List) 2817
20.1.4.2.28 tcBdr (Table Cell Borders) 2818
20.1.4.2.29 tcStyle (Table Cell Style) 2819
20.1.4.2.30 tcTxStyle (Table Cell Text Style) 2819
20.1.4.2.31 tl2br (Top Left to Bottom Right Border) 2820
20.1.4.2.32 top (Top Border) 2820
20.1.4.2.33 tr2bl (Top Right to Bottom Left Border) 2821
20.1.4.2.34 wholeTbl (Whole Table) 2821

20.1.5 3D 2821
20.1.5.1 anchor (Anchor Point) 2822
20.1.5.2 backdrop (Backdrop Plane) 2822
20.1.5.3 bevelB (Bottom Bevel) 2823
20.1.5.4 bevelT (Top Bevel) 2824
20.1.5.5 camera (Camera) 2825
20.1.5.6 contourClr (Contour Color) 2827
20.1.5.7 extrusionClr (Extrusion Color) 2828
20.1.5.8 flatTx (No text in 3D scene) 2829
20.1.5.9 lightRig (Light Rig) 2829
20.1.5.10 norm (Normal) 2830
20.1.5.11 rot (Rotation) 2831

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.5.12 sp3d (Apply 3D shape properties) 2832
20.1.5.13 up (Up Vector) 2834

20.1.6 Shared Style Sheet 2835

20.1.6.1 clrMap (Color Map) 2836
20.1.6.2 clrScheme (Color Scheme) 2837
20.1.6.3 custClrLst (Custom Color List) 2839
20.1.6.4 extraClrScheme (Extra Color Scheme) 2839
20.1.6.5 extraClrSchemeLst (Extra Color Scheme List) 2840
20.1.6.6 masterClrMapping (Master Color Mapping) 2841
20.1.6.7 objectDefaults (Object Defaults) 2841
20.1.6.8 overrideClrMapping (Override Color Mapping) 2841
20.1.6.9 theme (Theme) 2843
20.1.6.10 themeElements (Theme Elements) 2843
20.1.6.11 themeManager (Theme Manager) 2844
20.1.6.12 themeOverride (Theme Override) 2844

20.1.7 Coordinate Systems and Transformations 2845

20.1.7.1 chExt (Child Extents) 2845
20.1.7.2 chOff (Child Offset) 2845
20.1.7.3 ext (Extents) 2846
20.1.7.4 off (Offset) 2847
20.1.7.5 xfrm (2D Transform for Grouped Objects) 2848
20.1.7.6 xfrm (2D Transform for Individual Objects) 2849

20.1.8 Shape Fills, Effects, and Line Properties 2850

20.1.8.1 alphaBiLevel (Alpha Bi-Level Effect) 2850
20.1.8.2 alphaCeiling (Alpha Ceiling Effect) 2850
20.1.8.3 alphaFloor (Alpha Floor Effect) 2850
20.1.8.4 alphaInv (Alpha Inverse Effect) 2850
20.1.8.5 alphaMod (Alpha Modulate Effect) 2851
20.1.8.6 alphaModFix (Alpha Modulate Fixed Effect) 2851
20.1.8.7 alphaOutset (Alpha Inset/Outset Effect) 2851
20.1.8.8 alphaRepl (Alpha Replace Effect) 2852
20.1.8.9 bevel (Line Join Bevel) 2852
20.1.8.10 bgClr (Background color) 2852
20.1.8.11 biLevel (Bi-Level (Black/White) Effect) 2852
20.1.8.12 blend (Blend Effect) 2853
20.1.8.13 blip (Blip) 2853
20.1.8.14 blipFill (Picture Fill) 2854
20.1.8.15 blur (Blur Effect) 2856
20.1.8.16 clrChange (Color Change Effect) 2857
20.1.8.17 clrFrom (Change Color From) 2858
20.1.8.18 clrRepl (Solid Color Replacement) 2858
20.1.8.19 clrTo (Change Color To) 2858
20.1.8.20 cont (Effect Container) 2858
20.1.8.21 custDash (Custom Dash) 2858
20.1.8.22 ds (Dash Stop) 2859
20.1.8.23 duotone (Duotone Effect) 2859

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.8.24 effect (Effect) 2859
20.1.8.25 effectDag (Effect Container) 2860
20.1.8.26 effectLst (Effect Container) 2860
20.1.8.27 fgClr (Foreground color) 2862
20.1.8.28 fill (Fill) 2863
20.1.8.29 fillOverlay (Fill Overlay Effect) 2863
20.1.8.30 fillRect (Fill Rectangle) 2863
20.1.8.31 fillToRect (Fill To Rectangle) 2864
20.1.8.32 glow (Glow Effect) 2866
20.1.8.33 gradFill (Gradient Fill) 2866
20.1.8.34 grayscl (Gray Scale Effect) 2868
20.1.8.35 grpFill (Group Fill) 2868
20.1.8.36 gs (Gradient stops) 2868
20.1.8.37 gsLst (Gradient Stop List) 2868
20.1.8.38 headEnd (Line Head/End Style) 2868
20.1.8.39 hsl (Hue Saturation Luminance Effect) 2869
20.1.8.40 innerShdw (Inner Shadow Effect) 2870
20.1.8.41 lin (Linear Gradient Fill) 2870
20.1.8.42 lum (Luminance Effect) 2871
20.1.8.43 miter (Miter Line Join) 2871
20.1.8.44 noFill (No Fill) 2872
20.1.8.45 outerShdw (Outer Shadow Effect) 2872
20.1.8.46 path (Path Gradient) 2873
20.1.8.47 pattFill (Pattern Fill) 2874
20.1.8.48 prstDash (Preset Dash) 2874
20.1.8.49 prstShdw (Preset Shadow) 2875
20.1.8.50 reflection (Reflection Effect) 2875
20.1.8.51 relOff (Relative Offset Effect) 2878
20.1.8.52 round (Round Line Join) 2878
20.1.8.53 softEdge (Soft Edge Effect) 2878
20.1.8.54 solidFill (Solid Fill) 2879
20.1.8.55 srcRect (Source Rectangle) 2879
20.1.8.56 stretch (Stretch) 2879
20.1.8.57 tailEnd (Tail line end style) 2880
20.1.8.58 tile (Tile) 2880
20.1.8.59 tileRect (Tile Rectangle) 2882
20.1.8.60 tint (Tint Effect) 2883
20.1.8.61 xfrm (Transform Effect) 2883

20.1.9 Shape Definitions and Attributes 2884

20.1.9.1 ahLst (List of Shape Adjust Handles) 2885
20.1.9.2 ahPolar (Polar Adjust Handle) 2885
20.1.9.3 ahXY (XY Adjust Handle) 2886
20.1.9.4 arcTo (Draw Arc To) 2887
20.1.9.5 avLst (List of Shape Adjust Values) 2889
20.1.9.6 close (Close Shape Path) 2890
20.1.9.7 cubicBezTo (Draw Cubic Bezier Curve To) 2891
20.1.9.8 custGeom (Custom Geometry) 2892

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.9.9 cxn (Shape Connection Site) 2893
20.1.9.10 cxnLst (List of Shape Connection Sites) 2895
20.1.9.11 gd (Shape Guide) 2895
20.1.9.12 gdLst (List of Shape Guides) 2898
20.1.9.13 lnTo (Draw Line To) 2898
20.1.9.14 moveTo (Move Path To) 2899
20.1.9.15 path (Shape Path) 2899
20.1.9.16 pathLst (List of Shape Paths) 2901
20.1.9.17 pos (Shape Position Coordinate) 2902
20.1.9.18 prstGeom (Preset geometry) 2904
20.1.9.19 prstTxWarp (Preset Text Warp) 2905
20.1.9.20 pt (Shape Path Point) 2908
20.1.9.21 quadBezTo (Draw Quadratic Bezier Curve To) 2910
20.1.9.22 rect (Shape Text Rectangle) 2910

20.1.10 Simple Types 2911

20.1.10.1 ST_AdjAngle (Adjustable Angle Methods) 2911
20.1.10.2 ST_AdjCoordinate (Adjustable Coordinate Methods) 2911
20.1.10.3 ST_Angle (Angle) 2912
20.1.10.4 ST_AnimationBuildType (Animation Build Type) 2912
20.1.10.5 ST_AnimationChartBuildType (Chart Animation Build Type) 2912
20.1.10.6 ST_AnimationChartOnlyBuildType (Chart only Animation Types) 2912
20.1.10.7 ST_AnimationDgmBuildType (Diagram Animation Build Type) 2913
20.1.10.8 ST_AnimationDgmOnlyBuildType (Diagram only Animation Types) 2913
20.1.10.9 ST_BevelPresetType (Bevel Presets) 2914
20.1.10.10 ST_BlackWhiteMode (Black and White Mode) 2918
20.1.10.11 ST_BlendMode (Blend Mode) 2919
20.1.10.12 ST_BlipCompression (Blip Compression Type) 2919
20.1.10.13 ST_ChartBuildStep (Chart Animation Build Step) 2919
20.1.10.14 ST_ColorSchemeIndex (Theme Color Reference) 2920
20.1.10.15 ST_CompoundLine (Compound Line Type) 2921
20.1.10.16 ST_Coordinate (Coordinate) 2921
20.1.10.17 ST_Coordinate32 (Coordinate Point) 2921
20.1.10.18 ST_Coordinate32Unqualified (Coordinate Point) 2922
20.1.10.19 ST_CoordinateUnqualified (Coordinate) 2922
20.1.10.20 ST_DgmBuildStep (Diagram Animation Build Steps) 2922
20.1.10.21 ST_DrawingElementId (Drawing Element ID) 2922
20.1.10.22 ST_EffectContainerType (Effect Container Type) 2923
20.1.10.23 ST_FixedAngle (Fixed Angle) 2923
20.1.10.24 ST_FixedPercentage (Fixed Percentage) 2924
20.1.10.25 ST_FontCollectionIndex (Font Collection Index) 2924
20.1.10.26 ST_FOVAngle (Field of View Angle) 2924
20.1.10.27 ST_GeomGuideFormula (Geometry Guide Formula Properties) 2924
20.1.10.28 ST_GeomGuideName (Geometry Guide Name Properties) 2925
20.1.10.29 ST_LightRigDirection (Light Rig Direction) 2925
20.1.10.30 ST_LightRigType (Light Rig Type) 2929
20.1.10.31 ST_LineCap (End Line Cap) 2938
20.1.10.32 ST_LineEndLength (Line End Length) 2938

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.10.33 ST_LineEndType (Line End Type) 2939
20.1.10.34 ST_LineEndWidth (Line End Width) 2939
20.1.10.35 ST_LineWidth (Line Width) 2940
20.1.10.36 ST_OnOffStyleType (On/Off Style Type) 2940
20.1.10.37 ST_PathFillMode (Path Fill Mode) 2941
20.1.10.38 ST_PathShadeType (Path Shade Type) 2941
20.1.10.39 ST_PenAlignment (Alignment Type) 2941
20.1.10.40 ST_Percentage (Percentage) 2942
20.1.10.41 ST_PitchFamily (Pitch Family) 2942
20.1.10.42 ST_PositiveCoordinate (Positive Coordinate) 2943
20.1.10.43 ST_PositiveCoordinate32 (Positive Coordinate Point) 2943
20.1.10.44 ST_PositiveFixedAngle (Positive Fixed Angle) 2944
20.1.10.45 ST_PositiveFixedPercentage (Positive Fixed Percentage) 2944
20.1.10.46 ST_PositivePercentage (Positive Percentage Value with Sign) 2944
20.1.10.47 ST_PresetCameraType (Preset Camera Type) 2944
20.1.10.48 ST_PresetColorVal (Preset Color Value) 2965
20.1.10.49 ST_PresetLineDashVal (Preset Line Dash Value) 2971
20.1.10.50 ST_PresetMaterialType (Preset Material Type) 2972
20.1.10.51 ST_PresetPatternVal (Preset Pattern Value) 2981
20.1.10.52 ST_PresetShadowVal (Preset Shadow Type) 2984
20.1.10.53 ST_RectAlignment (Rectangle Alignments) 2987
20.1.10.54 ST_SchemeColorVal (Scheme Color) 2987
20.1.10.55 ST_ShapeID (Shape ID) 2988
20.1.10.56 ST_ShapeType (Preset Shape Types) 2988
20.1.10.57 ST_StyleMatrixColumnIndex (Style Matrix Column Index) 3057
20.1.10.58 ST_SystemColorVal (System Color Value) 3057
20.1.10.59 ST_TextAlignType (Text Alignment Types) 3059
20.1.10.60 ST_TextAnchoringType (Text Anchoring Types) 3060
20.1.10.61 ST_TextAutonumberScheme (Text Auto-number Schemes) 3060
20.1.10.62 ST_TextBulletSizePercent (Bullet Size Percentage) 3063
20.1.10.63 ST_TextBulletStartAtNum (Start Bullet At Number) 3063
20.1.10.64 ST_TextCapsType (Text Cap Types) 3063
20.1.10.65 ST_TextColumnCount (Text Column Count) 3064
20.1.10.66 ST_TextFontAlignType (Font Alignment Types) 3064
20.1.10.67 ST_TextFontScalePercentOrPercentString (Text Font Scale Percentage) 3065
20.1.10.68 ST_TextFontSize (Text Font Size) 3065
20.1.10.69 ST_TextHorzOverflowType (Text Horizontal Overflow Types) 3065
20.1.10.70 ST_TextIndent (Text Indentation) 3065
20.1.10.71 ST_TextIndentLevelType (Text Indent Level Type) 3066
20.1.10.72 ST_TextMargin (Text Margin) 3066
20.1.10.73 ST_TextNonNegativePoint (Text Non-Negative Point) 3066
20.1.10.74 ST_TextPoint (Text Point) 3067
20.1.10.75 ST_TextPointUnqualified (Text Point) 3067
20.1.10.76 ST_TextShapeType (Preset Text Shape Types) 3067
20.1.10.77 ST_TextSpacingPercentOrPercentString (Text Spacing Percent) 3081
20.1.10.78 ST_TextSpacingPoint (Text Spacing Point) 3081
20.1.10.79 ST_TextStrikeType (Text Strike Type) 3082
20.1.10.80 ST_TextTabAlignType (Text Tab Alignment Types) 3082

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.10.81 ST_TextTypeface (Text Typeface) 3082
20.1.10.82 ST_TextUnderlineType (Text Underline Types) 3083
20.1.10.83 ST_TextVerticalType (Vertical Text Types) 3084
20.1.10.84 ST_TextVertOverflowType (Text Vertical Overflow) 3085
20.1.10.85 ST_TextWrappingType (Text Wrapping Types) 3085
20.1.10.86 ST_TileFlipMode (Tile Flip Mode) 3085
20.1.10.87 ST_TextBulletSize (Bullet Size Percentage) 3087

End of informative text.

20.1.2 Basics

This section describes all the basic common elements associated with the DrawingML framework.

20.1.2.1 EMU Unit of Measurement

Throughout ISO/IEC 29500, the EMU is used as a unit of measurement for length. An *EMU* is defined as follows:

$$1 \text{ emu} = \frac{1}{914400} \text{US inch} = \frac{1}{360000} \text{cm}$$

[*Rationale*: The EMU was created in order to be able to evenly divide in both English and Metric units, in order to avoid rounding errors during the calculation. The usage of EMUs also facilitates a more seamless system switch and interoperability between different locales utilizing different units of measurement. EMUs define an integer based, high precision coordinate system. *end rationale*]

20.1.2.2 Core Drawing Object Information

Within DrawingML, there is the notion of core drawing elements. These are elements that both are vital to and common across the DrawingML framework. These elements denote the most integral pieces of the DrawingML document structure and thus are among the most widely used.

[*Note*: Measurement Units - Length units must be expressed in device-independent physical units: English Metric units (EMUs), points, picas, and inches. Device-dependent units such as pixels must not be used. *end note*]

20.1.2.2.1 bldChart (Build Chart)

This element specifies how to build the animation for a diagram.

[*Example*: Consider the following example where a chart is specified to be animated by category rather than as one entity. Thus, the bldChart element should be used as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:bdldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldSub>
      <a:bldChart bld="category"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bdldLst>
```

end example]

Attributes	Description
animBg (Animate Background)	Specifies whether or not the chart background elements should be animated as well. [*Note*: An example of background elements are grid lines and the chart legend. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bld (Build)	Specifies how the chart is built. The animation animates the sub-elements in the container in the particular order defined by this attribute. The possible values for this attribute are defined by the ST_AnimationChartBuildType simple type (§20.1.10.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationChartBuildProperties) is located in §A.4.1. *end note*]

20.1.2.2.2 bldDgm (Build Diagram)

This element specifies how to build the animation for a diagram.

[*Example*: Consider having a diagram appear as on entity as opposed to by section. The bldDgm element should be used as follows:

```
<p:bdldLst>
  <p:bldGraphic spid="4" grpId="0">
    <p:bldSub>
      <a:bldDgm bld="one"/>
    </p:bldSub>
  </p:bldGraphic>
</p:bdldLst>
```

end example]

Attributes	Description
bld (Build)	Specifies how the chart is built. The animation animates the sub-elements in the container in the particular order defined by this attribute.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_AnimationDgmBuildType simple type (§20.1.10.7).
rev (Reverse Animation)	Specifies whether the animation of the objects in this diagram should be reversed or not. If this attribute is not specified, a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationDgmBuildProperties) is located in §A.4.1. *end note*]

20.1.2.2.3 chart (Chart to Animate)

This element specifies a reference to a chart that should be animated within a sequence of slide animations. In addition to simply acting as a reference to a chart there is also animation build steps defined.

Attributes	Description
bldStep (Animation Build Step)	Specifies which step this part of the chart should be built using. For instance the chart can be built as one object meaning it is animated as a single graphic. Alternatively the chart can be animated, or built as separate pieces. The possible values for this attribute are defined by the ST_ChartBuildStep simple type (§20.1.10.13).
categoryIdx (Category Index)	Specifies the index of the category within the corresponding chart that should be animated. The possible values for this attribute are defined by the W3C XML Schema int datatype.
seriesIdx (Series Index)	Specifies the index of the series within the corresponding chart that should be animated. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationChartElement) is located in §A.4.1. *end note*]

20.1.2.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual drawing properties for a connector shape. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.2.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.1.2.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

20.1.2.2.7 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
preferRelativeResize (Relative Resize Preferred)	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

20.1.2.2.8 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<p:pic>
  …
  <p:nvPicPr>
    <p:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </p:nvPicPr>
  …
</p:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object)	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: <… descr="A picture of a bowl of fruit"> The descr attribute contains alternative text which can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden)	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier)	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name)	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title)	Specifies the title (caption) of the current DrawingML object.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.1.2.2.9 cNvSpPr (Non-Visual Shape Drawing Properties)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with

[*Example*: Consider the shape that has a shape lock applied to it.

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="2" name="Rectangle 1"/>
    <p:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </p:cNvSpPr>
  </p:nvSpPr>
…
</p:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box)	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

20.1.2.2.10 cxnSp (Connection Shape)

This element specifies a connection shape that is used to connect two sp elements. Once a connection is specified using a cxnSp, it is left to the generating application to determine the exact path the connector takes. That is the connector routing algorithm is left up to the generating application as the desired path might be different depending on the specific needs of the application.

[*Example*: Consider the following connector shape that connects two regular shapes.

```
<p:spTree>
  …
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="1" name="Rectangle 1"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:sp>
    <p:nvSpPr>
      <p:cNvPr id="2" name="Rectangle 2"/>
      <p:cNvSpPr/>
      <p:nvPr/>
    </p:nvSpPr>
    …
  </p:sp>
  <p:cxnSp>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <p:nvCxnSpPr>
        <p:cNvPr id="3" name="Elbow Connector 3"/>
        <p:cNvCxnSpPr>
          <a:stCxn id="1" idx="3"/>
          <a:endCxn id="2" idx="1"/>
        </p:cNvCxnSpPr>
        <p:nvPr/>
      </p:nvCxnSpPr>
      …
    </p:cxnSp>
  </p:spTree>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlConnector) is located in §A.4.1. *end note*]

20.1.2.2.11 cxnSpLocks (Connection Shape Locks)

This element specifies all locking properties for a connection shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorLocking) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.2.2.12 dgm (Diagram to Animate)

This element specifies a reference to a diagram that should be animated within a sequence of slide animations. In addition to simply acting as a reference to a diagram there is also animation build steps defined.

Attributes	Description
bldStep (Animation Build Step)	Specifies which step this part of the diagram should be built using. For instance the diagram can be built as one object meaning it is animated as a single graphic. Alternatively the diagram can be animated, or built as separate pieces. The possible values for this attribute are defined by the ST_DgmBuildStep simple type (§20.1.10.20).
id (Identifier)	Specifies the GUID of the shape for this build step in the animation. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimationDgmElement) is located in §A.4.1. *end note*]

20.1.2.2.13 endCxn (Connection End)

This element specifies the ending connection that should be made by the corresponding connector shape. This connects the end tail of the connector to the final destination shape.

Attributes	Description
id (Identifier)	Specifies the id of the shape to make the final connection to. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
idx (Index)	Specifies the index into the connection site table of the final connection shape. That is there are many connection sites on a shape and it shall be specified which connection site the corresponding connector shape should connect to. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connection) is located in §A.4.1. *end note*]

20.1.2.2.14 ext (Extension)

This element specifies an extension that is used for future extensions to the current version of DrawingML. This allows for the specifying of currently unknown elements in the future that is used for later versions of generating applications.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of this tag. The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtension) is located in §A.4.1. *end note*]

20.1.2.2.15 extLst (Extension List)

This element specifies the extension list within which all future extensions of element type ext is defined. The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the framework.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtensionList) is located in §A.4.1. *end note*]

20.1.2.2.16 graphic (Graphic Object)

This element specifies the existence of a single graphic object. Document authors should refer to this element when they wish to persist a graphical object of some kind. The specification for this graphical object is provided entirely by the document author and referenced within the graphicData child element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObject) is located in §A.4.1. *end note*]

20.1.2.2.17 graphicData (Graphic Object Data)

This element specifies the reference to a graphic object within the document. This graphic object is provided entirely by the document authors who choose to persist this data within the document.

[*Note*: Depending on the kind of graphical object used not every generating application that supports the OOXML framework has the ability to render the graphical object. *end note*]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	this tag. The possible values for this attribute are defined by the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectData) is located in §A.4.1. *end note*]

20.1.2.2.18 graphicFrame (Graphic Frame)

This element specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGraphicalObjectFrame) is located in §A.4.1. *end note*]

20.1.2.2.19 graphicFrameLocks (Graphic Frame Locks)

This element specifies all locking properties for a graphic frame. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding graphic frame. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noDrilldown (Disallow Selection of Child Shapes)	Specifies that the generating application should not allow selecting of objects within the corresponding graphic frame but allow selecting of the graphic frame itself. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding graphic frame. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the corresponding graphic frame cannot be moved. Objects that reside within the graphic frame can still be moved unless they also have been locked. If this attribute is not specified, then a value of false is assumed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding graphic frame. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding picture. That means also that no picture, shapes or text attached to this picture can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. [*Note*: If this attribute is specified to be true then the graphic frame cannot be selected and the objects within the graphic frame cannot be selected as well. That is the entire graphic frame including all sub-parts are considered un-selectable. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrameLocking) is located in §A.4.1. *end note*]

20.1.2.2.20 grpSp (Group shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<p:grpSp>
  <p:nvGrpSpPr>
    <p:cNvPr id="10" name="Group 9"/>
    <p:cNvGrpSpPr/>
    <p:nvPr/>
  </p:nvGrpSpPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:grpSpPr>
  <a:xfrm>
    <a:off x="838200" y="990600"/>
    <a:ext cx="2426208" cy="978408"/>
    <a:chOff x="838200" y="990600"/>
    <a:chExt cx="2426208" cy="978408"/>
  </a:xfrm>
</p:grpSpPr>
<p:sp>
…
</p:sp>
<p:sp>
…
</p:sp>
<p:sp>
…
</p:sp>
</p:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShape) is located in §A.4.1. *end note*]

20.1.2.2.21 grpSpLocks (Group Shape Locks)

This element specifies all locking properties for a connection shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the corresponding group shape cannot be grouped. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow	Specifies that the corresponding graphic frame cannot be moved. Objects that reside

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Moving Shape)	within the graphic frame can still be moved unless they also have been locked. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resizing)	Specifies that the corresponding group shape cannot be resized. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the corresponding group shape cannot be rotated Objects that reside within the group can still be rotated unless they also have been locked. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the corresponding group shape cannot have any part of it be selected. That means that no picture, shapes or attached text can be selected either if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. [*Note*: This property is inherited by sub-elements and thus all shapes within the group shape cannot be selected when this attribute is set to a value of true. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noUngrp (Disallow Shape Ungrouping)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupLocking) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.2.2.22 grpSpPr (Visual Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and	Specifies that the group shape should be rendered using only black and white coloring.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
White Mode)	That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

20.1.2.2.23 hlinkHover (Hyperlink for Hover)

This element specifies the hyperlink information to be activated when the user's mouse is hovered over the corresponding object. The operation of the hyperlink is to have the specified action be activated when the mouse of the user hovers over the object. When this action is activated then additional attributes can be used to specify other tasks that should be performed along with the action.

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of 1 or `true` is assumed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlink) is located in §A.4.1. *end note*]

20.1.2.2.24 ln (Outline)

This element specifies an outline style that can be applied to a number of different objects such as shapes and text. The line allows for the specifying of many different types of outlines including even line dashes and bevels.

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

20.1.2.2.25 nvCxnSpPr (Non-Visual Properties for a Connection Shape)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlConnectorNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.26 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGraphicFrameNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.27 nvGrpSpPr (Non-Visual Properties for a Group Shape)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShapeNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.28 nvPicPr (Non-Visual Properties for a Picture)

This element specifies all non-visual properties for a picture. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following PresentationML.

```
<p:pic>
  …
  <p:nvPicPr>
    …
  </p:nvPicPr>
  …
</p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlPictureNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.29 nvSpPr (Non-Visual Properties for a Shape)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlShapeNonVisual) is located in §A.4.1. *end note*]

20.1.2.2.30 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following PresentationML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <p:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <p:nvPr/>
  </p:nvPicPr>
  <p:blipFill>
  …
  </p:blipFill>
  <p:spPr>
  …
  </p:spPr>
</p:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlPicture) is located in §A.4.1. *end note*]

20.1.2.2.31 picLocks (Picture Locks)

This element specifies all locking properties for a graphic frame. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noCrop (Disallow Crop Changes)	Specifies that the generating application should not allow cropping for the corresponding picture. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureLocking) is located in §A.4.1. *end note*]

20.1.2.2.32 snd (Hyperlink Sound)

This element specifies a sound to be played when a hyperlink within the document is activated. This sound is specified from within the parent hyperlink element.

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name)	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

20.1.2.2.33 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the DrawingML framework. In addition to a geometry each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

[*Note*: Shapes are the preferred mechanism for specifying text on a slide. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlShape) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.2.2.34 spLocks (Shape Locks)

This element specifies all locking properties for a shape. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow Shape Grouping)	Specifies that the generating application should not allow shape grouping for the corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noTextEdit (Disallow Shape Text Editing)	Specifies that the generating application should not allow editing of the shape text for the corresponding shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeLocking) is located in §A.4.1. *end note*]

20.1.2.2.35 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a shape.

Attributes	Description
bwMode (Black and White Mode)	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	[*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.1.2.2.36 stCxn (Connection Start)

This element specifies the starting connection that should be made by the corresponding connector shape. This connects the head of the connector to the first shape.

Attributes	Description
id (Identifier)	Specifies the id of the shape to make the final connection to. The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
idx (Index)	Specifies the index into the connection site table of the final connection shape. That is there are many connection sites on a shape and it shall be specified which connection site the corresponding connector shape should connect to. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connection) is located in §A.4.1. *end note*]

20.1.2.2.37 style (Shape Style)

This element specifies the style information for a shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

20.1.2.2.38 sx (Horizontal Ratio)

This element specifies the horizontal ratio for use within a scaling calculation.

Attributes	Description
d (Denominator)	Specifies the denominator to be used within the equation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema long datatype.
n (Numerator)	Specifies the numerator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ratio) is located in §A.4.1. *end note*]

20.1.2.2.39 sy (Vertical Ratio)

This element specifies the vertical ratio for use within a scaling calculation.

Attributes	Description
d (Denominator)	Specifies the denominator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.
n (Numerator)	Specifies the numerator to be used within the equation. The possible values for this attribute are defined by the W3C XML Schema long datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Ratio) is located in §A.4.1. *end note*]

20.1.2.2.40 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

20.1.2.2.41 txSp (Text Shape)

This element specifies the existence of a text shape within a parent shape. This text shape is specifically used for displaying text as it has only text related child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlTextShape) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.2.2.42 useSpRect (Use Shape Text Rectangle)

This element specifies that the text rectangle from the parent shape should be used for this text shape. If this attribute is specified then the text rectangle, or text bounding box as it is also called should have the same dimensions as the text bounding box of the parent shape within which this text shape resides.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlUseShapeRectangle) is located in §A.4.1. *end note*]

20.1.2.2.43 cpLocks (Content Part Locks)

This element specifies all locking properties for a content part. These properties inform the generating application about specific properties that have been previously locked and thus should not be changed.

Attributes	Description
noAdjustHandles (Disallow Showing Adjust Handles)	Specifies that the generating application should not show adjust handles for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeArrowheads (Disallow Arrowhead Changes)	Specifies that the generating application should not allow arrowhead changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeAspect (Disallow Aspect Ratio Change)	Specifies that the generating application should not allow aspect ratio changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noChangeShapeType (Disallow Shape Type Change)	Specifies that the generating application should not allow shape type changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noEditPoints (Disallow Shape Point Editing)	Specifies that the generating application should not allow shape point changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noGrp (Disallow	Specifies that the generating application should not allow shape grouping for the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Shape Grouping)	corresponding connection shape. That is it cannot be combined within other shapes to form a group of shapes. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noMove (Disallow Shape Movement)	Specifies that the generating application should not allow position changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noResize (Disallow Shape Resize)	Specifies that the generating application should not allow size changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noRot (Disallow Shape Rotation)	Specifies that the generating application should not allow shape rotation changes for the corresponding connection shape. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
noSelect (Disallow Shape Selection)	Specifies that the generating application should not allow selecting of the corresponding connection shape. That means also that no picture, shapes, or text attached to this connection shape can be selected if this attribute has been specified. If this attribute is not specified, then a value of false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ContentPartLocking) is located in §A.4.1. *end note*]

20.1.2.3 Colors

Given its own section within DrawingML Basics, colors are an integral part of the DrawingML framework. Colors are used in virtually every object to help describe it's appearance when it is rendered on the screen. Since not every generating application wishes to represent color in the same manner, it is possible to specify color in a number of different ways.

20.1.2.3.1 alpha (Alpha)

This element specifies its input color with the specific opacity, but with its color unchanged.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage value.
	
	[*Example*: The following represents a green solid fill which is 50% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alpha val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.2 alphaMod (Alpha Modulation)

This element specifies a more or less opaque version of its input color. An alpha modulate never increases the alpha beyond 100%. A 200% alpha modulate makes an input color twice as opaque as before. A 50% alpha modulate makes an input color half as opaque as before.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage relative to the input color.
	
	[*Example*: The following represents a green solid fill which is 50% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alphaMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositivePercentage) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.2.3.3 alphaOff (Alpha Offset)

This element specifies a more or less opaque version of its input color. Increases or decreases the input alpha percentage by the specified percentage offset. A 10% alpha offset increases a 50% opacity to 60%. A -10% alpha offset decreases a 50% opacity to 40%. The transformed alpha values are limited to a range of 0 to 100%. A 10% alpha offset increase to a 100% opaque object still results in 100% opacity.

Attributes	Description

val (Value)	Specifies the opacity as expressed by a percentage offset increase or decrease to the input color. Increases never increase the opacity beyond 100%, decreases never decrease the opacity below 0%.
	
	[*Example*: The following represents a green solid fill which is 90% opaque
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:alphaOff val="-10%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.4 blue (Blue)

This element specifies the input color with the specific blue component, but with the red and green color components unchanged.

Attributes	Description

val (Value)	Specifies the value of the blue component. The assigned value is specified as a percentage with 0% indicating minimal blue and 100% indicating maximum blue.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, FF, FF)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:blue val="100%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.5 blueMod (Blue Modulation)

This element specifies the input color with its blue component modulated by the given percentage. A 50% blue modulate reduces the blue component by half. A 200% blue modulate doubles the blue component.

Attributes	Description

val (Value)	Specifies the blue component as expressed by a percentage relative to the input color component. Increases never increase the blue component beyond 100%, decreases never decrease the blue component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, 00, 80)
	
	`<a:solidFill>`
	`  <a:srgbClr val="0000FF">`
	`  <a:blueMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.6 blueOff (Blue Offset)

This element specifies the input color with its blue component shifted, but with its red and green color components unchanged.

Attributes	Description
val (Value)	Specifies the blue component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the blue component beyond 100%, decreases never decrease the blue component below 0%.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, 00, CC)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:blueOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.7 comp (Complement)

This element specifies that the color rendered should be the complement of its input color with the complement being defined as such. Two colors are called complementary if, when mixed they produce a shade of grey. For instance, the complement of red which is RGB (255, 0, 0) is cyan which is RGB (0, 255, 255).

Primary colors and secondary colors are typically paired in this way:

- red and cyan (where cyan is the mixture of green and blue)
- green and magenta (where magenta is the mixture of red and blue)
- blue and yellow (where yellow is the mixture of red and green)

[*Example*:

The following represents the complement of red:

```
<a:solidFill>
  <a:srgbClr val="FF0000">
    <a:comp/>
  </a:srgbClr>
</a:solidFill>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ComplementTransform) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.2.3.8 gamma (Gamma)

This element specifies that the output color rendered by the generating application should be the sRGB gamma shift of the input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GammaTransform) is located in §A.4.1. *end note*]

20.1.2.3.9 gray (Gray)

This element specifies a grayscale of its input color, taking into relative intensities of the red, green, and blue primaries.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GrayscaleTransform) is located in §A.4.1. *end note*]

20.1.2.3.10 green (Green)

This elements specifies the input color with the specified green component, but with its red and blue color components unchanged.

Attributes	Description

val (Value)	Specifies the value of the green component. The assigned value is specified as a percentage with 0% indicating minimal green and 100% indicating maximum green.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, 00, FF) to value RRGGBB= (00, FF, FF)
	
	`<a:solidFill>`
	`  <a:srgbClr val="0000FF">`
	`  <a:green val="100%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.11 greenMod (Green Modulation)

This element specifies the input color with its green component modulated by the given percentage. A 50% green modulate reduces the green component by half. A 200% green modulate doubles the green component.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

val (Value)	Specifies the green component as expressed by a percentage relative to the input color component. Increases never increase the green component beyond 100%, decreases never decrease the green component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 80, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:greenMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.12 greenOff (Green Offset)

This element specifies the input color with its green component shifted, but with its red and blue color components unchanged.

Attributes	Description

val (Value)	Specifies the green component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the green component beyond 100%, decreases never decrease the green component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, CC, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:greenOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.13 hslClr (Hue, Saturation, Luminance Color Model)

This element specifies a color using the HSL color model. A perceptual gamma of 2.2 is assumed.

Hue refers to the dominant wavelength of color, saturation refers to the purity of its hue, and luminance refers to its lightness or darkness.

As with all colors, colors defined with the HSL color model can have color transforms applied to it.

[*Example*:

The color blue having RGB value RRGGBB = (00, 00, 80) is equivalent to

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
```

end example]

Attributes	Description
hue (Hue)	Specifies the angular value describing the wavelength. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
lum (Luminance)	Specifies the luminance referring to the lightness or darkness of the color. Expressed as a percentage with 0% referring to maximal dark (black) and 100% referring to maximal white. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sat (Saturation)	Specifies the saturation referring to the purity of the hue. Expressed as a percentage with 0% referring to grey, 100% referring to the purest form of the hue. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HslColor) is located in §A.4.1. *end note*]

20.1.2.3.14 hue (Hue)

This element specifies the input color with the specified hue, but with its saturation and luminance unchanged.

[*Example*: The following two solid fills are equivalent.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="0" sat="100%" lum="50%">
    <a:hue val="14400000"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the actual angle value to be used with the input color's hue component. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedAngle) is located in §A.4.1. *end note*]

20.1.2.3.15 hueMod (Hue Modulate)

This element specifies the input color with its hue modulated by the given percentage. A 50% hue modulate decreases the angular hue value by half. A 200% hue modulate doubles the angular hue value.

Attributes	Description

val (Value)	Specifies the hue as expressed by a percentage relative to the input color.
	
	[*Example*: The following manipulates the fill color from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB = (FF, FF, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:hueMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositivePercentage) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.2.3.16 hueOff (Hue Offset)

This element specifies the input color with its hue shifted, but with its saturation and luminance unchanged.

Attributes	Description

val (Value)	Specifies the actual angular value of the shift. The result of the shift shall be between 0 and 360 degrees. Shifts resulting in angular values less than 0 are treated as 0. Shifts resulting in angular values greater than 360 are treated as 360.
	
	[*Example*:
	The following increases the hue angular value by 10 degrees.
	
	`<a:solidFill>`
	`  <a:hslClr hue="0" sat="100%" lum="50%"/>`
	`  <a:hueOff val="600000"/>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Angle) is located in §A.4.1. *end note*]

20.1.2.3.17 inv (Inverse)

This element specifies the inverse of its input color.

[*Example*:

The inverse of red (1, 0, 0) is cyan (0, 1, 1).

The following represents cyan, the inverse of red:

```
<a:solidFill>
  <a:srgbClr val="FF0000">
    <a:inv/>
  </a:srgbClr>
</a:solidFill>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_InverseTransform) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.2.3.18 invGamma (Inverse Gamma)

This element specifies that the output color rendered by the generating application should be the inverse sRGB gamma shift of the input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_InverseGammaTransform) is located in §A.4.1. *end note*]

20.1.2.3.19 lum (Luminance)

This element specifies the input color with the specified luminance, but with its hue and saturation unchanged. Typically luminance values fall in the range [0%, 100%].

[*Example*:

The following two solid fills are equivalent:

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="0%">
    <a:lum val="50%"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description

val (Value)	Specifies the value of the luminance. The assigned value is specified as a percentage with 0% indicating minimal luminance and 100% indicating maximum luminance.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 66, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lum val="20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.20 lumMod (Luminance Modulation)

This element specifies the input color with its luminance modulated by the given percentage. A 50% luminance modulate reduces the luminance by half. A 200% luminance modulate doubles the luminance.

Attributes	Description

val (Value)	Specifies the luminance as expressed by a percentage relative to the input color. Increases never increase the luminance beyond 100%, decreases never decrease the luminance below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 75, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lumMod val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.21 lumOff (Luminance Offset)

This element specifies the input color with its luminance shifted, but with its hue and saturation unchanged.

Attributes	Description

val (Value)	Specifies the luminance as expressed by a percentage offset increase or decrease to the input color. Increases never increase the luminance beyond 100%, decreases never decrease the luminance below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, 99, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:lumOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.22 prstClr (Preset Color)

This element specifies a color which is bound to one of a predefined collection of colors.

[*Example*:

The following defines a solid fill bound to the "black" preset color.

```
<a:solidFill>
  <a:prstClr val="black">
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the actual preset color value. The possible values for this attribute are defined by the ST_PresetColorVal simple type (§20.1.10.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetColor) is located in §A.4.1. *end note*]

20.1.2.3.23 red (Red)

This element specifies the input color with the specified red component, but with its green and blue color components unchanged.

Attributes	Description
val (Value)	Specifies the value of the red component. The assigned value is specified as a percentage with 0% indicating minimal red and 100% indicating maximum red. [*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (FF, FF, 00)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:solidFill> <a:srgbClr val="00FF00"> <a:red val="100%"/> </a:srgbClr> </a:solidFill> *end example*] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.24 redMod (Red Modulation)

This element specifies the input color with its red component modulated by the given percentage. A 50% red modulate reduces the red component by half. A 200% red modulate doubles the red component.

Attributes	Description
val (Value)	Specifies the red component as expressed by a percentage relative to the input color component. Increases never increase the red component beyond 100%, decreases never decrease the red component below 0%. [*Example*: The following manipulates the fill from having RGB value RRGGBB = (FF, 00, 00) to value RRGGBB= (80, 00, 00) <a:solidFill> <a:srgbClr val="FF0000"> <a:redMod val="50%"/> </a:srgbClr> </a:solidFill> *end example*] The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.25 redOff (Red Offset)

This element specifies the input color with its red component shifted, but with its green and blue color components unchanged.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

val (Value)	Specifies the red component as expressed by a percentage offset increase or decrease to the input color component. Increases never increase the red component beyond 100%, decreases never decrease the red component below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (FF, 00, 00) to value RRGGBB= (CC, 00, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="FF0000">`
	`  <a:redOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.26 sat (Saturation)

This element specifies the input color with the specified saturation, but with its hue and luminance unchanged. Typically saturation values fall in the range [0%, 100%].

[*Example*:

The following two solid fills are equivalent:

```
<a:solidFill>
  <a:hslClr hue="14400000" sat="100%" lum="50%">
</a:solidFill>
<a:solidFill>
  <a:hslClr hue="14400000" sat="0%" lum="50%">
    <a:sat val="100000"/>
  <a:hslClr/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	Specifies the value of the saturation. The assigned value is specified as a percentage with 0% indicating minimal saturation and 100% indicating maximum saturation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (40, C0, 40)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:sat val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.27 satMod (Saturation Modulation)

This element specifies the input color with its saturation modulated by the given percentage. A 50% saturation modulate reduces the saturation by half. A 200% saturation modulate doubles the saturation.

Attributes	Description

val (Value)	Specifies the saturation as expressed by a percentage relative to the input color. Increases never increase the saturation beyond 100%, decreases never decrease the saturation below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (66, 99, 66)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:satMod val="20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.2.3.28 satOff (Saturation Offset)

This element specifies the input color with its saturation shifted, but with its hue and luminance unchanged. A 10% offset to 20% saturation yields 30% saturation.

Attributes	Description

val (Value)	Specifies the saturation as expressed by a percentage offset increase or decrease to the input color. Increases never increase the saturation beyond 100%, decreases never decrease the saturation below 0%.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (19, E5, 19)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:satOff val="-20%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Percentage) is located in §A.4.1. *end note*]

20.1.2.3.29 schemeClr (Scheme Color)

This element specifies a color bound to a user's theme. As with all elements which define a color, it is possible to apply a list of color transforms to the base color defined.

Attributes	Description

val (Value)	Specifies the desired scheme.
	
	[*Example*: The following represents a color bound to the "lt1" theme color
	
	`<a:solidFill>`
	`  <a:schemeClr val="lt1"/>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_SchemeColorVal simple type (§20.1.10.54).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SchemeColor) is located in §A.4.1. *end note*]

20.1.2.3.30 scrgbClr (RGB Color Model - Percentage Variant)

This element specifies a color using the red, green, blue RGB color model. Each component, red, green, and blue is expressed as a percentage from 0% to 100%. A linear gamma of 1.0 is assumed.

Specifies the level of red as expressed by a percentage offset increase or decrease relative to the input color.

[*Example*: The following represent the same color

```
<a:solidFill>
  <a:scrgbClr r="50%" g="50%" b="50%"/>
</a:solidFill>
<a:solidFill>
  <a:srgbClr val="BCBCBC"/>
</a:solidFill>
```

end example]

Attributes	Description
b (Blue)	Specifies the percentage of blue. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
g (Green)	Specifies the percentage of green. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Red)	Specifies the percentage of red. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScRgbColor) is located in §A.4.1. *end note*]

20.1.2.3.31 shade (Shade)

This element specifies a darker version of its input color. A 10% shade is 10% of the input color combined with 90% black.

Attributes	Description
val (Value)	Specifies the shade as expressed by a percentage value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (00, BC, 00)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:shade val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.2.3.32 srgbClr (RGB Color Model - Hex Variant)

This element specifies a color using the red, green, blue RGB color model. Red, green, and blue is expressed as sequence of hex digits, RRGGBB. A perceptual gamma of 2.2 is used.

Specifies the level of red as expressed by a percentage offset increase or decrease relative to the input color.

[*Example*: The following represent the same color

```
<a:solidFill>
  <a:scrgbClr r="50%" g="50%" b="50%"/>
</a:solidFill>
<a:solidFill>
  <a:srgbClr val="BCBCBC"/>
</a:solidFill>
```

end example]

Attributes	Description
val (Value)	The actual color value. Expressed as a sequence of hex digits RRGGBB. The possible values for this attribute are defined by the ST_HexColorRGB simple type (§22.9.2.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SRgbColor) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.2.3.33 sysClr (System Color)

This element specifies a color bound to predefined operating system elements.

[*Example*: The following represents the default color used for displaying text in a window.

```
<a:solidFill>
  <a:sysClr val="windowText"/>
</a:solidFill>
```

end example]

Attributes	Description
lastClr (Last Color)	Specifies the color value that was last computed by the generating application. The possible values for this attribute are defined by the ST_HexColorRGB simple type (§22.9.2.5).
val (Value)	Specifies the system color value. The possible values for this attribute are defined by the ST_SystemColorVal simple type (§20.1.10.58).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SystemColor) is located in §A.4.1. *end note*]

20.1.2.3.34 tint (Tint)

This element specifies a lighter version of its input color. A 10% tint is 10% of the input color combined with 90% white.

Attributes	Description

val (Value)	Specifies the tint as expressed by a percentage value.
	
	[*Example*: The following manipulates the fill from having RGB value RRGGBB = (00, FF, 00) to value RRGGBB= (BC, FF, BC)
	
	`<a:solidFill>`
	`  <a:srgbClr val="00FF00">`
	`  <a:tint val="50%"/>`
	`  </a:srgbClr>`
	`</a:solidFill>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.3 Audio and Video

The Audio and Video portion of the DrawingML framework deals with all media of these two kinds that can be attached to objects within a document. Types of audio that can be represented within a file are CD audio, QuickTime audio, and any other generic audio. When dealing with generic audio there is the option for embedding it within the file and also linking it. The linking option is preferable if the size of the audio file is too large and thus increases the size of the document by an undesirable amount. For video there are two kinds that can be represented and that is either a QuickTime movie or any other generic movie. When dealing with generic video there is only the option of linking to the media as video is too large to embed within a document.

20.1.3.1 audioCd (Audio from CD)

This element specifies the existence of Audio from a CD. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the sound however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has an audio from a CD attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:audioCd>
        <a:st track="1"/>
        <a:end track="3" time="65"/>
      </a:audioCd>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single audioCD element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this audioCD element from within the timing node list. For this example we see that the audio for this CD starts playing at the 0 second mark on the first track and ends on the 1 minute 5 second mark of the third track. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCD) is located in §A.4.1. *end note*]

20.1.3.2 audioFile (Audio from File)

This element specifies the existence of an audio file. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the audio however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has an audio file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:audioFile r:link="rId1"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single audioFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this audioFile element from within the timing node list. The Linked relationship id is used to retrieve the actual audio file for playback purposes. *end example*]

Attributes	Description
contentType (Content Type of Linked Audio File)	Specifies the content type for the external file that is referenced by this element. Content types define a media type, a subtype, and an optional set of parameters, as defined in Part 2. If a rendering application cannot process external content of the content type specified, then the specified content can be ignored. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] If this attribute is omitted, application should attempt to determine the content type by reading the contents of the relationship's target. A producer that wants interoperability should use the following standard format: • audio/mpeg ISO/IEC 11172-3 The possible values for this attribute are defined by the W3C XML Schema string

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for a linked object. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioFile) is located in §A.4.1. *end note*]

20.1.3.3 end (Audio End Time)

This element specifies the end point for a CD Audio sound element. Encompassed within this element are the time and track at which the sound should halt its playback. This element is used in conjunction with an Audio Start Time element to specify the time span for an entire audioCD sound element.

[*Example*: Consider the following DrawingML.

```
<a:audioCd>
  <a:st track="1" time="2"/>
  <a:end track="3" time="65"/>
</a:audioCd>
```

In the above example, the audioCD sound element shown specifies for a portion of audio spanning from 2 seconds into the first track to 1 minute, 5 seconds into the third track. *end example*]

Attributes	Description
time (Time)	Specifies the time in seconds that the CD Audio should be stopped at. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
track (Track)	Specifies which track of the CD this Audio stops playing at. This attribute is required and cannot be omitted. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCDTime) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.3.4 quickTimeFile (QuickTime from File)

This element specifies the existence of a QuickTime file, as defined in the 2007-09-04 version of the QuickTime File Format Specification: http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf. [*Note*: For more information on the QuickTime format: http://developer.apple.com/reference/QuickTime/. *end note*]. This element is specified within the non-visual properties of an object. The QuickTime file shall be attached to an object as this is how it is represented within the document. The actual playing of the QuickTime however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has a QuickTime file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:quickTimeFile r:link="rId1"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single quickTimeFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this quickTimeFile element from within the timing node list. The Linked relationship id is used to retrieve the actual video file for playback purposes. *end example*]

Attributes	Description
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for a linked object. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_QuickTimeFile) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.3.5 st (Audio Start Time)

This element specifies the start point for a CD Audio sound element. Encompassed within this element are the time and track at which the sound should begin its playback. This element is used in conjunction with an Audio End Time element to specify the time span for an entire audioCD sound element.

[*Example*: Consider the following DrawingML.

```
<a:audioCd>
  <a:st track="1" time="2"/>
  <a:end track="3" time="65"/>
</a:audioCd>
```

In the above example, the audioCD sound element shown specifies for a portion of audio spanning from 2 seconds into the first track to 1 minute, 5 seconds into the third track. *end example*]

Attributes	Description
time (Time)	Specifies the time in seconds that the CD Audio should be started at. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
track (Track)	Specifies which track of the CD this Audio begins playing on. This attribute is required and cannot be omitted. The possible values for this attribute are defined by the W3C XML Schema unsignedByte datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AudioCDTime) is located in §A.4.1. *end note*]

20.1.3.6 videoFile (Video from File)

This element specifies the existence of a video file. This element is specified within the non-visual properties of an object. The video shall be attached to an object as this is how it is represented within the document. The actual playing of the video however is done within the timing node list that is specified under the timing element.

[*Example*: Consider the following picture object that has a video attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <p:cNvPicPr>
        <a:picLocks noRot="1"/>
      </p:cNvPicPr>
      <p:nvPr>
        <a:videoFile r:link="rId1"/>
      </p:nvPr>
    </p:nvPicPr>
    …
  </p:pic>
```

In the above example, we see that there is a single videoFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this videoFile element from within the timing node list. The Linked relationship id is used to retrieve the actual video file for playback purposes. *end example*]

Attributes	Description
contentType (Content Type of Linked Video File)	Specifies the content type for the external file that is referenced by this element. Content types define a media type, a subtype, and an optional set of parameters, as defined in Part 2. If a rendering application cannot process external content of the content type specified, then the specified content can be ignored. [*Note*: A list of suggested video types is provided in §15.2.17. *end note*] If this attribute is omitted, application should attempt to determine the content type by reading the contents of the relationship's target. The possible values for this attribute are defined by the W3C XML Schema string datatype.
link (Linked Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for a linked video file. This attribute is used to specify the location of an object that does not reside within this file. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_VideoFile) is located in §A.4.1. *end note*]

20.1.3.7 wavAudioFile (Audio from WAV File)

This element specifies the existence of an audio WAV file. This element is specified within the non-visual properties of an object. The audio shall be attached to an object as this is how it is represented within the document. The actual playing of the audio however is done within the timing node list that is specified under the timing element.

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following picture object that has an audio WAV file attached to it.

```
<p:pic>
  <p:nvPicPr>
    <p:cNvPr id="7" name="Rectangle 6">
      <a:hlinkClick r:id="" action="ppaction://media"/>
    </p:cNvPr>
    <p:cNvPicPr>
      <a:picLocks noRot="1"/>
    </p:cNvPicPr>
    <p:nvPr>
      <a:wavAudioFile r:embed="rId2"/>
    </p:nvPr>
  </p:nvPicPr>
  …
</p:pic>
```

In the above example, we see that there is a single wavAudioFile element attached to this picture. This picture is placed within the document just as a normal picture or shape would be. The id of this picture, namely 7 in this case, is used to refer to this wavAudioFile element from within the timing node list. The Embedded relationship id is used to retrieve the actual audio file for playback purposes. *end example*]

[*Note*: This element is generally used for the purposes of embedding audio files within the document. For linking to generic audio files the audioFile element should be used. *end note*]

Attributes	Description
embed (Embedded Audio File Relationship ID) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for an embedded audio file. This attribute is used to specify the location of an object that resides locally within the file. [*Note*: A list of suggested audio types is provided in §15.2.2. *end note*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
name (Sound Name)	Specifies the original name or given short name for the corresponding sound. This is used to distinguish this sound from others by providing a human readable name for the attached sound should the user need to identify the sound among others within the UI. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmbeddedWAVAudioFile) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.4 Styles

Styles within DrawingML refer to the way a particular object (be it text or a shape, or anything else) is formatted. Different aspects, ranging from color, line type, fill, and effects applied to the object can be predefined within a theme. The main purpose of a theme is to define a style matrix from which a document can pull style information from in order to format the visual look of objects in a document.

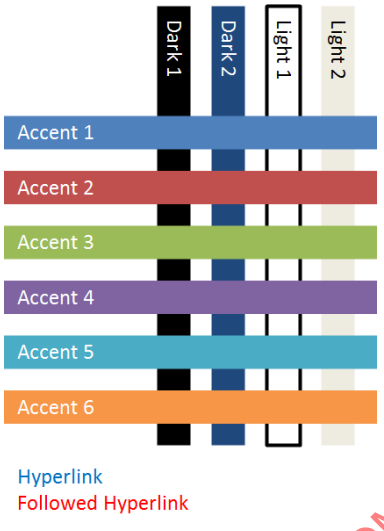
20.1.4.1 Styles

The elements in this section compose the basic definition of a style, including its associated colors, effect styles, line styles, fill styles, background styles, and font scheme.

20.1.4.1.1 accent1 (Accent 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

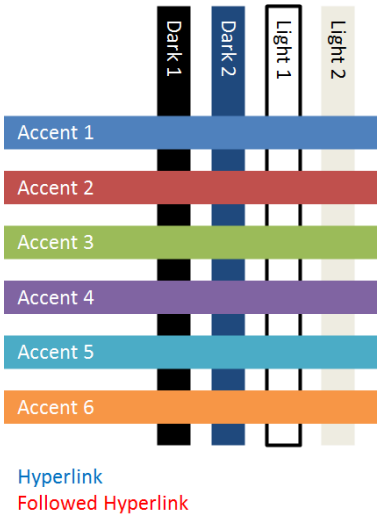
20.1.4.1.2 accent2 (Accent 2)

This element defines a color that happens to be the accent 2 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



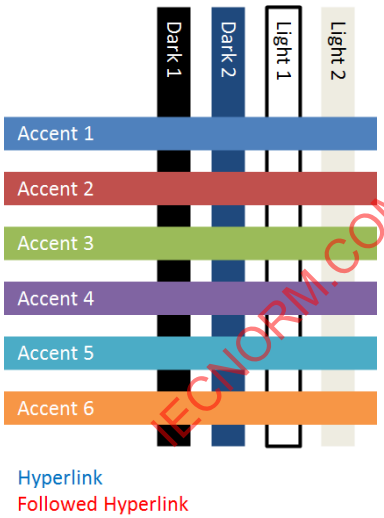
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.3 accent3 (Accent 3)

This element defines a color that happens to be the accent 3 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

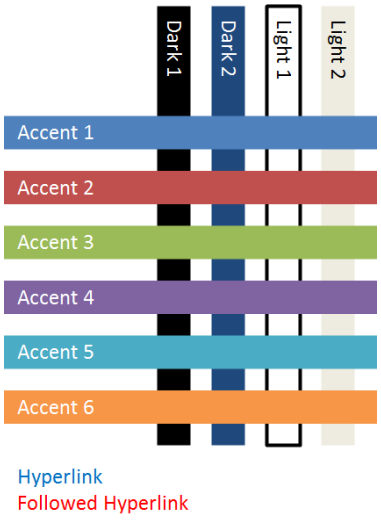
[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.4.1.4 accent4 (Accent 4)

This element defines a color that happens to be the accent 4 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.5 accent5 (Accent 5)

This element defines a color that happens to be the accent 5 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

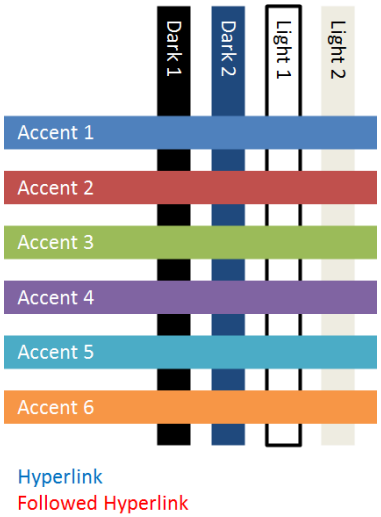
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.6 accent6 (Accent 6)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.7 bgFillStyleLst (Background Fill Style List)

This element defines a list of background fills that are used within a theme. The background fills consist of three fills, arranged in order from subtle to moderate to intense.

[*Example*: Consider the following example of a background fill style list within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<bgFillStyleLst>
  <solidFill>
…
  </solidFill>
  <gradFill rotWithShape="1">
…
  </gradFill>
  <blipFill>
…
  </blipFill>
</bgFillStyleLst>
```

In this example, we see that the list contains a solid fill for the subtle fill, a gradient fill for the moderate fill and an image fill for the intense background fill. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BackgroundFillStyleList) is located in §A.4.1. *end note*]

20.1.4.1.8 custClr (Custom color)

This element defines a custom color. The custom colors are used within a custom color list to define custom colors that are extra colors that can be appended to a theme. This is useful within corporate scenarios where there is a set corporate color palette from which to work.

Attributes	Description
name (Name)	The name of the color shown in the color picker. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomColor) is located in §A.4.1. *end note*]

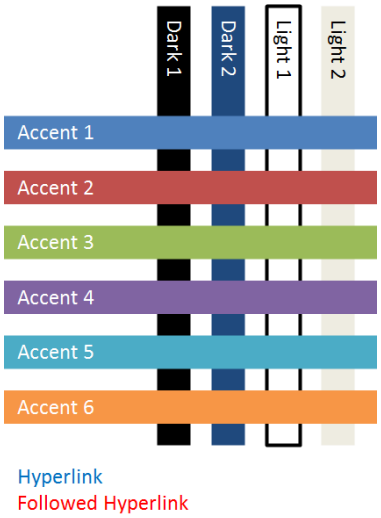
20.1.4.1.9 dk1 (Dark 1)

This element defines a color that happens to be the dark 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



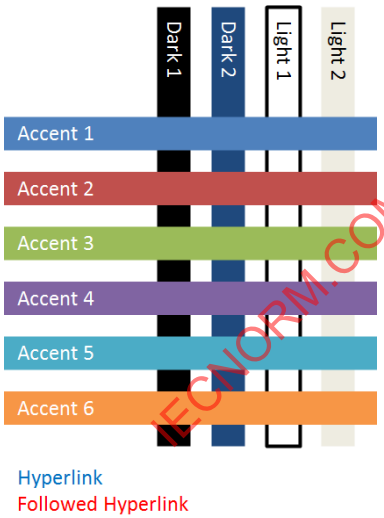
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.10 dk2 (Dark 2)

This element defines a color that happens to be the dark 2 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.4.1.11 effectStyle (Effect Style)

This element defines a set of effects and 3D properties that can be applied to an object.

[*Example*: Consider the following example of an effect style within DrawingML:

```
<effectStyle>
  <effectLst>
    <outerShdw blurRad="57150" dist="38100" dir="5400000" algn="ctr"
      rotWithShape="0">
      <schemeClr val="phClr">
        <shade val="9000"/>
        <satMod val="105000"/>
        <alpha val="48000"/>
      </schemeClr>
    </outerShdw>
  </effectLst>
</effectStyle>
```

In this example, an outer shadow is being applied to a shape as the moderate effect. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectStyleItem) is located in §A.4.1. *end note*]

20.1.4.1.12 effectStyleLst (Effect Style List)

This element defines a set of three effect styles that create the effect style list for a theme. The effect styles are arranged in order of subtle to moderate to intense.

[*Example*: Consider the following example of an effect style list within DrawingML:

```
<effectStyleLst>
  <effectStyle>
    <effectLst>
      <outerShdw blurRad="57150" dist="38100" dir="5400000"
        algn="ctr" rotWithShape="0">
…
      </outerShdw>
    </effectLst>
  </effectStyle>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
  <effectStyle>
    <effectLst>
      <outerShdw blurRad="57150" dist="38100" dir="5400000"
        algn="ctr" rotWithShape="0">
…
      </outerShdw>
    </effectLst>
  </effectStyle>
  <effectStyle>
    <effectLst>
      <outerShdw blurRad="57150" dist="38100" dir="5400000"
        algn="ctr" rotWithShape="0">
…
      </outerShdw>
    </effectLst>
    <scene3d>
…
    </scene3d>
    <sp3d prstMaterial="powder">
…
    </sp3d>
  </effectStyle>
</effectStyleLst>
```

In this example, we see three effect styles defined. The first two (subtle and moderate) define an outer shadow as the effect, while the third effect style (intense) defines an outer shadow along with 3D properties which are to be applied to the object as well. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectStyleList) is located in §A.4.1. *end note*]

20.1.4.1.13 fillStyleLst (Fill Style List)

This element defines a set of three fill styles that are used within a theme. The three fill styles are arranged in order from subtle to moderate to intense.

[*Example*: Consider the following example of a fill style list within DrawingML:

```
<fillStyleLst>
  <solidFill>
…
  </solidFill>
  <gradFill rotWithShape="1">
…
  </gradFill>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <gradFill rotWithShape="1">
  …
    </gradFill>
  </fillStyleLst>
```

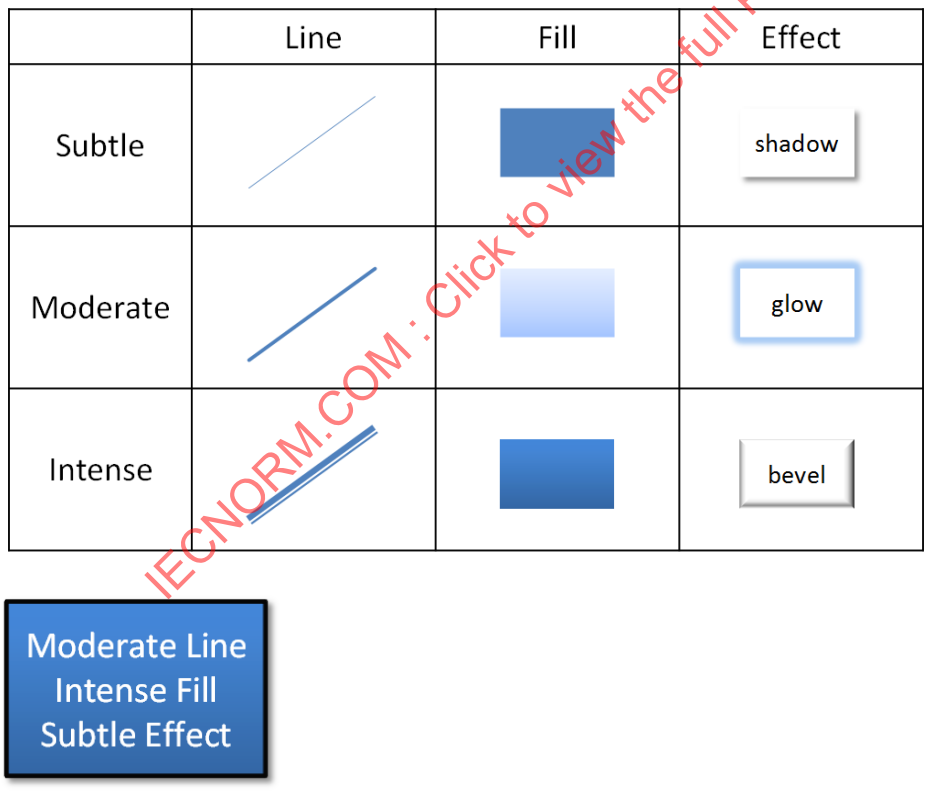
In this example, we see three fill styles being defined within the fill style list. The first style is the subtle style and defines simply a solid fill. The second and third styles (moderate and intense fills respectively) define gradient fills. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillStyleList) is located in §A.4.1. *end note*]

20.1.4.1.14 fmtScheme (Format Scheme)

This element contains the background fill styles, effect styles, fill styles, and line styles which define the style matrix for a theme. The style matrix consists of subtle, moderate, and intense fills, lines, and effects. The background fills are not generally thought of to directly be associated with the matrix, but do play a role in the style of the overall document. Usually, a given object chooses a single line style, a single fill style, and a single effect style in order to define the overall final look of the object.

[*Example*: Consider the following example of the style matrix in use within DrawingML:



In this example, we see a shape styled which utilizes different aspects from the above defined style matrix. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

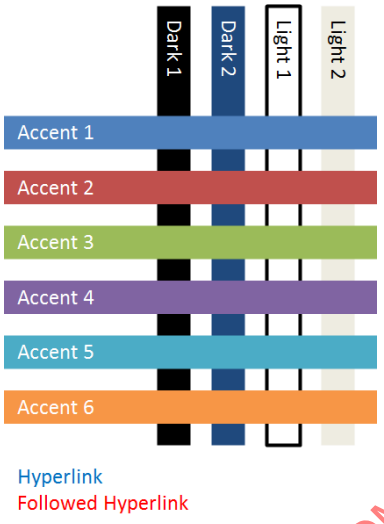
Attributes	Description
name (Name)	Defines the name for the format scheme. The name is simply a human readable string which identifies the format scheme in the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrix) is located in §A.4.1. *end note*]

20.1.4.1.15 folHlink (Followed Hyperlink)

This element defines a color that happens to be the followed hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.16 font (Font)

This element defines a font within the styles area of DrawingML. A font is defined by a script along with a typeface.

[*Example*: Consider the following example of a font in DrawingML:

```
<font script="Thai" typeface="Cordia New"/>
```

In this example, we see that the script 'Thai' is supposed to use the font face 'Cordia New'. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
script (Script)	Specifies the script, or language, in which the typeface is supposed to be used. [*Note*: It is recommended that script names as specified in ISO 15924 are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
typeface (Typeface)	Specifies the font face to use. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SupplementalFont) is located in §A.4.1. *end note*]

20.1.4.1.17 fontRef (Font Reference)

This element represents a reference to a themed font. When used it specifies which themed font to use along with a choice of color.

[*Example*: Consider the following example of a font reference within DrawingML:

```
<fontRef idx="minor">
  <schemeClr val="tx1"/>
</fontRef>
```

In this example, we see a font referencing the minor font defined within the theme. *end example*]

Attributes	Description
idx (Identifier)	Specifies the identifier of the font to reference. The possible values for this attribute are defined by the ST_FontCollectionIndex simple type (§20.1.10.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontReference) is located in §A.4.1. *end note*]

20.1.4.1.18 fontScheme (Font Scheme)

This element defines the font scheme within the theme. The font scheme consists of a pair of major and minor fonts for which to use in a document. The major font corresponds well with the heading areas of a document, and the minor font corresponds well with the normal text or paragraph areas.

[*Example*: Consider the following example of a font scheme within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<fontScheme name="sample">
  <majorFont>
…
  </majorFont>
  <minorFont>
…
  </minorFont>
</fontScheme>
```

In this example, we see the major and minor font lists within the font scheme that is named 'sample'. *end example*]

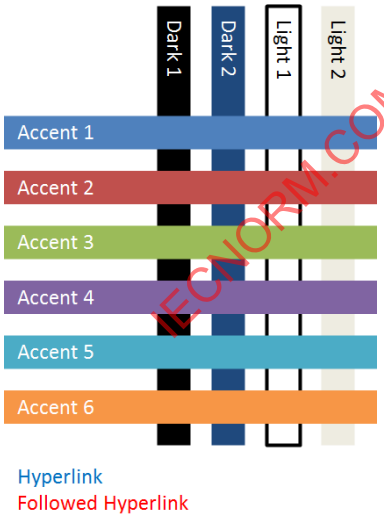
Attributes	Description
name (Name)	The name of the font scheme shown in the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontScheme) is located in §A.4.1. *end note*]

20.1.4.1.19 hlink (Hyperlink)

This element defines a color that happens to be the hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.20 lnDef (Line Default)

This element defines a default line that is used within a document.

[*Example*: Consider the following example of a default line defined in DrawingML:

```
<lnDef>
  <spPr/>
  <bodyPr/>
  <lstStyle/>
  <style>
    <lnRef idx="1">
      <schemeClr val="accent2"/>
    </lnRef>
    <fillRef idx="0">
      <schemeClr val="accent2"/>
    </fillRef>
    <effectRef idx="0">
      <schemeClr val="accent2"/>
    </effectRef>
    <fontRef idx="minor">
      <schemeClr val="tx1"/>
    </fontRef>
  </style>
</lnDef>
```

In this example, we see that the default line for the document is being defined as a themed line which references the subtle line style with idx equal to 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.1.21 lnStyleLst (Line Style List)

This element defines a list of three line styles for use within a theme. The three line styles are arranged in order from subtle to moderate to intense versions of lines. This list makes up part of the style matrix.

[*Example*: Consider the following example of a line style list within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<lnStyleLst>
  <ln w="9525" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr">
        <shade val="50000"/>
        <satMod val="103000"/>
      </schemeClr>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
  <ln w="25400" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr"/>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
  <ln w="38100" cap="flat" cmpd="sng" algn="ctr">
    <solidFill>
      <schemeClr val="phClr"/>
    </solidFill>
    <prstDash val="solid"/>
  </ln>
</lnStyleLst>
```

In this example, we see three lines defined within a line style list. The first line corresponds to the subtle line, the second to the moderate, and the third corresponds to the intense line defined in the theme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineStyleList) is located in §A.4.1. *end note*]

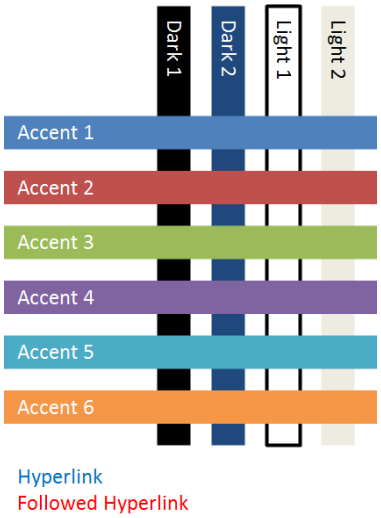
20.1.4.1.22 lt1 (Light 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



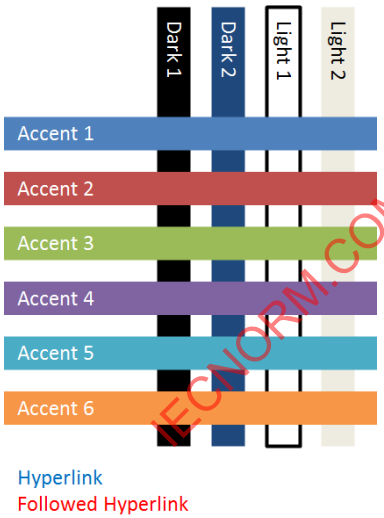
end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.4.1.23 lt2 (Light 2)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[*Example*: Consider the following example of a set of colors that form a color scheme:



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.4.1.24 majorFont (Major Font)

This element defines the set of major fonts which are to be used under different languages or locals.

[*Example*: Consider the following example of the major fonts being defined within DrawingML:

```
<majorFont>
  <latin typeface="Calibri"/>
  <ea typeface="Arial"/>
  <cs typeface="Arial"/>
  <font script="Jpan" typeface="ＭＳ Ｐゴシック"/>
  <font script="Hang" typeface="HY중고딕"/>
  <font script="Hans" typeface="隶书"/>
  <font script="Hant" typeface="微軟正黑體"/>
  <font script="Arab" typeface="Traditional Arabic"/>
  <font script="Hebr" typeface="Arial"/>
  <font script="Thai" typeface="Cordia New"/>
  <font script="Ethi" typeface="Nyala"/>
  <font script="Beng" typeface="Vrinda"/>
  <font script="Gujr" typeface="Shruti"/>
  <font script="Khmr" typeface="DaunPenh"/>
  <font script="Knda" typeface="Tunga"/>
</majorFont>
```

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.1.25 minorFont (Minor fonts)

This element defines the set of minor fonts that are to be used under different languages or locals.

[*Example*: Consider the following example of the minor fonts being defined within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<minorFont>
  <latin typeface="Calibri"/>
  <ea typeface="Arial"/>
  <cs typeface="Arial"/>
  <font script="Jpan" typeface="MS Pゴシック"/>
  <font script="Hang" typeface="HY중고딕"/>
  <font script="Hans" typeface="隶书"/>
  <font script="Hant" typeface="微軟正黑體/>
  <font script="Arab" typeface="Traditional Arabic"/>
  <font script="Hebr" typeface="Arial"/>
  <font script="Thai" typeface="Cordia New"/>
  <font script="Ethi" typeface="Nyala"/>
  <font script="Beng" typeface="Vrinda"/>
  <font script="Gujr" typeface="Shruti"/>
  <font script="Khmr" typeface="DaunPenh"/>
  <font script="Knda" typeface="Tunga"/>
</minorFont>
```

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.1.26 scene3d (3D Scene Properties)

This element defines optional scene-level 3D properties to apply to an object.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scene3D) is located in §A.4.1. *end note*]

20.1.4.1.27 spDef (Shape Default)

This element defines the formatting that is associated with the default shape. The default formatting can be applied to a shape when it is initially inserted into a document.

[*Example*: Consider the following example of a shape default being used within DrawingML:

```
<spDef>
  <spPr>
    <solidFill>
      <schemeClr val="accent2">
        <shade val="75000"/>
      </schemeClr>
    </solidFill>
  </spPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <bodyPr rtlCol="0" anchor="ctr"/>
    <lstStyle>
      <defPPr algn="ctr">
        <defRPr/>
      </defPPr>
    </lstStyle>
    <style>
      <lnRef idx="1">
        <schemeClr val="accent1"/>
      </lnRef>
      <fillRef idx="2">
        <schemeClr val="accent1"/>
      </fillRef>
      <effectRef idx="1">
        <schemeClr val="accent1"/>
      </effectRef>
      <fontRef idx="minor">
        <schemeClr val="dk1"/>
      </fontRef>
    </style>
  </spDef>
```

In this example, we see a default shape which references a certain themed fill, line, effect, and font along with an override fill to these. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.1.28 txDef (Text Default)

This element defines the default formatting which is applied to text in a document by default. The default formatting can and should be applied to the shape when it is initially inserted into a document.

[*Example*: Consider the following example of a text default being used within DrawingML:

```
  <txDef>
    <spPr>
      <solidFill>
        <schemeClr val="accent2">
          <shade val="75000"/>
        </schemeClr>
      </solidFill>
    </spPr>
    <bodyPr rtlCol="0" anchor="ctr"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <lstStyle>
      <defPPr algn="ctr">
        <defRPr/>
      </defPPr>
    </lstStyle>
    <style>
      <lnRef idx="1">
        <schemeClr val="accent1"/>
      </lnRef>
      <fillRef idx="2">
        <schemeClr val="accent1"/>
      </fillRef>
      <effectRef idx="1">
        <schemeClr val="accent1"/>
      </effectRef>
      <fontRef idx="minor">
        <schemeClr val="dk1"/>
      </fontRef>
    </style>
  </txDef>
```

In this example, we see a default text which references a certain themed fill, line, effect, and font along with an override fill to these. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DefaultShapeDefinition) is located in §A.4.1. *end note*]

20.1.4.2 Table Styles

Table styles are responsible for the rapid formatting that can be applied to a table. This rapid formatting takes different parts of a table into account, such as if the first row or last row should be emphasized, or if there is some type of banding (row for example) present on the table. All of these different types of formatting can be defined within a table style.

20.1.4.2.1 band1H (Band 1 Horizontal)

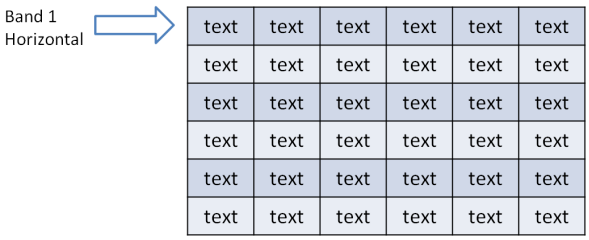
This element describes the formatting for the first row in horizontal banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 1 horizontal being used within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<band1H>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band1H>
```



In this example, we set the fill to be a solid fill referencing the accent 1 color defined in the theme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.2 band1V (Band 1 Vertical)

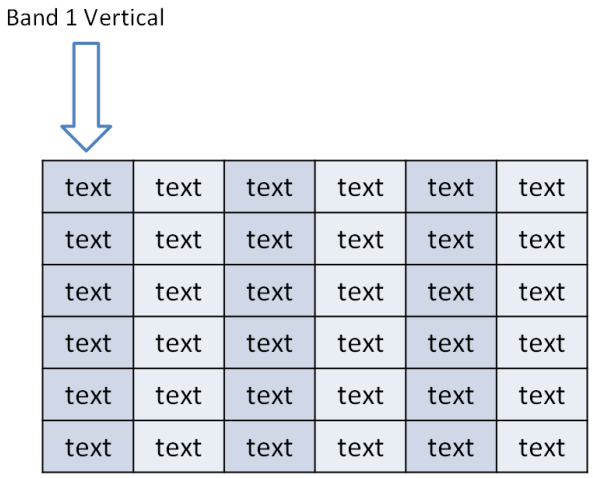
This element describes the formatting for the first row in vertical banding. Two different column formattings are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 1 vertical being used within DrawingML:

```
<band1V>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band1V>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



This markup example sets the fill to be a solid fill referencing the accent 1 color defined in the theme. *end example*]

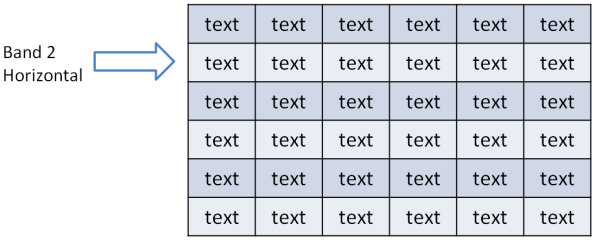
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.3 band2H (Band 2 Horizontal)

This element describes the formatting for the second row in horizontal banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 2 horizontal being used within DrawingML:

```
<band2H>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent2">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band2H>
```



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example, we set the fill to be a solid fill referencing the accent 2 color defined in the theme. *end example*]

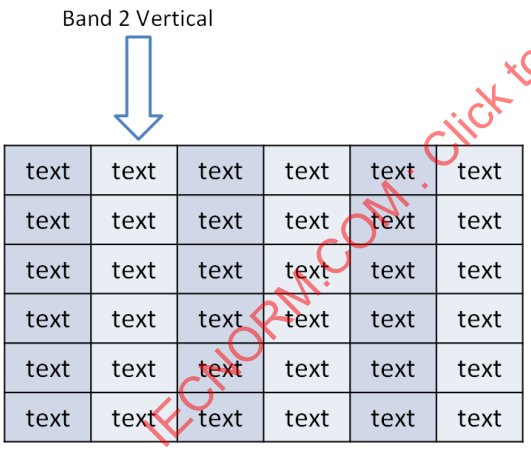
[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.4 band2V (Band 2 Vertical)

This element describes the formatting for the second row in vertical banding. Two different row formatting are applied to the table alternating in order to create a banding effect on the table.

[*Example*: Consider the following example of band 2 vertical being used within DrawingML:

```
<band2V>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent2">
          <tint val="40000"/>
        </schemeClr>
      </solidFill>
    </fill>
  </tcStyle>
</band2V>
```



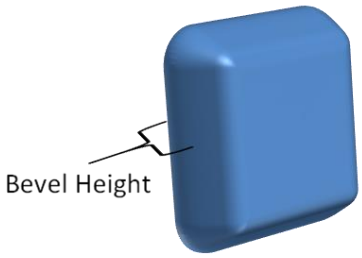
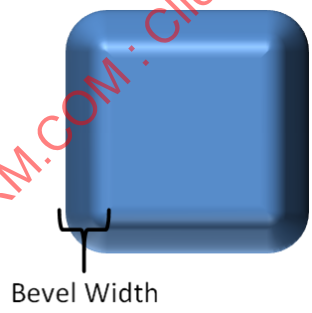
In this example, we set the fill to be a solid fill referencing the accent 2 color defined in the theme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.4.2.5 bevel (Bevel)

This element defines the properties of the bevel associated with the 3D effect applied to a cell in a table.

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the height of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.4.2.6 bottom (Bottom Border)

This element defines the line properties associated with the bottom border in a table cell.

[*Example*: Consider the following example of the bottom border in use within DrawingML:

```
<bottom>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</bottom>
```

In this example, we see the bottom border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.7 effect (Effect)

This element defines the effect that can be applied to a table as a whole through a table style.

[*Example*: Consider the following example of an effect in use within DrawingML:

```
<effect>
  <effectLst>
    <glow rad="228600">
      <schemeClr val="accent1">
        <satMod val="175000"/>
        <alpha val="40000"/>
      </schemeClr>
    </glow>
  </effectLst>
</effect>
```

In this example, we see a glow being defined within the table style that is applied to the table as a whole. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectProperties) is located in §A.4.1. *end note*]

20.1.4.2.8 effectRef (Effect Reference)

This element defines a reference to an effect style within the style matrix. The idx attribute refers the index of an effect style within the effectStyleLst element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

20.1.4.2.9 fill (Fill)

This element defines the fill that is applied to the table as a whole. The background of the table can contain a single fill that is the entire size of the table. This can allow for gradient fills, or image fills, which span the entire size of the table.

[*Example*: Consider the following example of a fill on a table background in DrawingML:

```
<fill>
  <gradFill flip="none" rotWithShape="1">
    <gsLst>
      <gs pos="0">
        <schemeClr val="accent2">
          <shade val="75000"/>
        </schemeClr>
      </gs>
      <gs pos="100000">
        <schemeClr val="accent2">
          <shade val="75000"/>
          <tint val="20000"/>
        </schemeClr>
      </gs>
    </gsLst>
    <lin ang="2700000" scaled="1"/>
    <tileRect/>
  </gradFill>
</fill>
```

In this example, we apply a gradient fill to the entire table on the background shape of the table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.4.2.10 fillRef (Fill Reference)

This element defines a reference to a fill style within the style matrix. The idx attribute refers to the index of a fill style or background fill style within the presentation's style matrix, defined by the fmtScheme element. A value of 0 or 1000 indicates no background, values 1-999 refer to the index of a fill style within the fillStyleLst element, and values 1001 and above refer to the index of a background fill style within the bgFillStyleLst element. The value 1001 corresponds to the first background fill style, 1002 to the second background fill style, and so on.

[*Example*:

```
<a:fillRef idx="2">
  <a:schemeClr val="accent2"/>
</a:fillRef>
```

The above code indicates the object is to have the style's second fill style using the accent2 color of the color scheme.

end example]

[*Example*:

```
<a:fillRef idx="1001">
  <a:schemeClr val="accent2"/>
</a:fillRef>
```

The above code indicates the object is to have the style's first background fill style using the accent2 color of the color scheme.

end example]

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

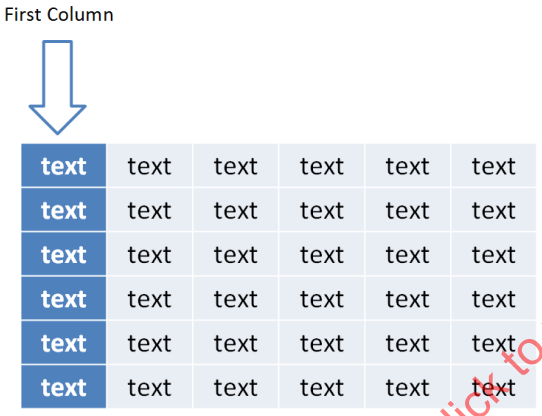
20.1.4.2.11 firstCol (First Column)

This element defines the cell formatting which can be applied to the first column of the table.

[*Example*: Consider the following example of first column formatting within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<firstCol>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</firstCol>
```



In this example, we define the first column cell fills to be accent 1 along with the text properties to be bold when first column formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.12 firstRow (First Row)

This element defines the cell formatting which can be applied to the first row of the table.

[*Example*: Consider the following example of first row formatting within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<firstRow>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</firstRow>
```

In this example, we define the first row cell fills to be accent 1 along with the text properties to be bold when first row formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.13 font (Font)

This element defines the font to be used within a given table cell text style. This element allows for exact definition of the font within the table style instead of referencing a themed font.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FontCollection) is located in §A.4.1. *end note*]

20.1.4.2.14 insideH (Inside Horizontal Border)

This element defines the line properties associated with the inner horizontal borders in a table.

[*Example*: Consider the following example of the inner horizontal borders in use within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<insideH>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</insideH>
```

In this example, we see the inner horizontal borders in a table to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.15 insideV (Inside Vertical Border)

This element defines the line properties associated with the inner vertical borders in a table.

[*Example*: Consider the following example of the inside vertical borders in use within DrawingML:

```
<insideV>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</insideV>
```

In this example, we see the inner vertical borders in a table to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.16 lastCol (Last Column)

This element defines the cell formatting which can be applied to the last column of the table.

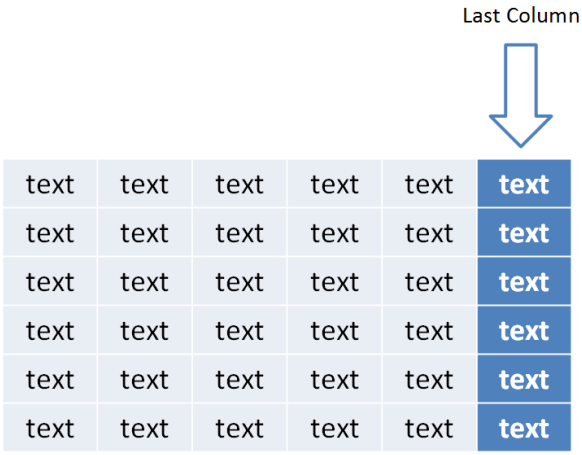
[*Example*: Consider the following example of last column formatting within DrawingML:

```
<lastCol>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <tcStyle>
      <tcBdr/>
      <fill>
        <solidFill>
          <schemeClr val="accent1"/>
        </solidFill>
      </fill>
    </tcStyle>
  </lastCol>
```



In this example, we define the last column cell fills to be accent 1 along with the text properties to be bold when last column formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.17 lastRow (Last Row)

This element defines the cell formatting which can be applied to the last row of the table.

[*Example*: Consider the following example of last row formatting within DrawingML:

```
  <lastRow>
    <tcTxStyle b="on">
      <fontRef idx="minor">
        <scrgbClr r="0" g="0" b="0"/>
      </fontRef>
      <schemeClr val="lt1"/>
    </tcTxStyle>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <tcStyle>
      <tcBdr/>
      <fill>
        <solidFill>
          <schemeClr val="accent1"/>
        </solidFill>
      </fill>
    </tcStyle>
  </lastRow>
```

In this example, we define the last row cell fills to be accent 1 along with the text properties to be bold when last row formatting is enabled through the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.18 left (Left Border)

This element defines the line properties associated with the left border in a table cell.

[*Example*: Consider the following example of the left border in use within DrawingML:

```
<left>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</left>
```

In this example, we see the left border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.4.2.19 lnRef (Line Reference)

This element defines a reference to a line style within the style matrix. The idx attribute refers the index of a line style within the fillStyleLst element.

Attributes	Description
idx (Style Matrix Index)	Specifies the style matrix index of the style referred to. The possible values for this attribute are defined by the ST_StyleMatrixColumnIndex simple type (§20.1.10.57).

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleMatrixReference) is located in §A.4.1. *end note*]

20.1.4.2.20 neCell (Northeast Cell)

This element defies the formatting for the cell in the northeast corner of a table when both the first row formatting and last column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the northeast cell formatting within DrawingML:

```
<neCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</neCell>
```

In this example, we specifically set the northeast cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.4.2.21 nwCell (Northwest Cell)

This element defies the formatting for the cell in the northwest corner of a table when both the first row formatting and first column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the northwest cell formatting within DrawingML:

```
<nwCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</nwCell>
```

In this example, we specifically set the northwest cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.22 right (Right Border)

This element defines the line properties associated with the right border in a table cell.

[*Example*: Consider the following example of the right border in use within DrawingML:

```
<right>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</right>
```

In this example, we see the right border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.23 seCell (Southeast Cell)

This element defies the formatting for the cell in the southeast corner of a table when both the last row formatting and last column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the southeast cell formatting within DrawingML:

```
<seCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</seCell>
```

In this example, we specifically set the southeast cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.24 swCell (Southwest Cell)

This element defies the formatting for the cell in the southwest corner of a table when both the last row formatting and first column formatting are enabled. This formatting is only applied to the single cell which overlaps between the two formatting options.

[*Example*: Consider the following example of the southwest cell formatting within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<swCell>
  <tcTxStyle b="on">
    <fontRef idx="minor">
      <scrgbClr r="0" g="0" b="0"/>
    </fontRef>
    <schemeClr val="lt1"/>
  </tcTxStyle>
  <tcStyle>
    <tcBdr/>
    <fill>
      <solidFill>
        <schemeClr val="accent1"/>
      </solidFill>
    </fill>
  </tcStyle>
</swCell>
```

In this example, we specifically set the southwest cell to contain bold text with a solid cell fill in the color of accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.4.2.25 tblBg (Table Background)

This element defines the formatting options which can be applied to the table background shape. The background shape is the same size as the entire table and can hold a fill or an effect which spans the entire table.

[*Example*: Consider the following example of a table background in use within DrawingML:

```
<tblBg>
  <fillRef idx="2">
    <schemeClr val="accent1"/>
  </fillRef>
  <effectRef idx="1">
    <schemeClr val="accent1"/>
  </effectRef>
</tblBg>
```

In this example, we see that there is a themed fill and themed effect being applied to the table background through the table style. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableBackgroundStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.4.2.26 tblStyle (Table Style)

This is the root element for a table style. Within the table style are different formatting options available in order to apply a table.

Attributes	Description
styleId (Style ID)	Specifies a GUID identifying the table style in a unique manner. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
styleName (Name)	Specifies the name of the table style which can show up in the user interface identifying the style to a user. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyle) is located in §A.4.1. *end note*]

20.1.4.2.27 tblStyleLst (Table Style List)

This element is simply a list of table styles which are used within a document.

[*Example*: Consider the following example of a table style list within DrawingML:

```
<tblStyleLst def="{5C22544A-7EE6-4342-B048-85BDC9FD1C3A}">
  <tblStyle styleId="{5C22544A-7EE6-4342-B048-85BDC9FD1C3A}"
    styleName="Medium Style 2 - Accent 1">
    …
  </tblStyle>
  <tblStyle styleId="{3C2FFA5D-87B4-456A-9821-1D502468CF0F}"
    styleName="Themed Style 1 - Accent 1">
    …
  </tblStyle>
</tblStyleLst>
```

In this example, we see two table styles defined along with the default being specified. *end example*]

Attributes	Description
def (Default)	The GUID corresponding to the default table style in the list of table styles. This default can be used when a table is initially inserted into a document. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleList) is located in §A.4.1. *end note*]

20.1.4.2.28 tcBdr (Table Cell Borders)

This element defines the borders for the cells within a table.

[*Example*: Consider the following example of table cell borders being used within DrawingML:

```
<tcBdr>
  <left>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </left>
  <right>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </right>
  <top>
    <lnRef idx="1">
      <schemeClr val="accent1"/>
    </lnRef>
  </top>
  <bottom>
    <lnRef idx="2">
      <schemeClr val="lt1"/>
    </lnRef>
  </bottom>
  <insideH>
    <ln>
      <noFill/>
    </ln>
  </insideH>
  <insideV>
    <ln>
      <noFill/>
    </ln>
  </insideV>
</tcBdr>
```

In this example, we define borders for the bottom, top, right, and left borders of the table cells. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCellBorderStyle) is located in §A.4.1. *end note*]

20.1.4.2.29 tcStyle (Table Cell Style)

This element defines the style for a give cell in a table.

[*Example*: Consider the following example of a table cell style in use within DrawingML:

```
<tcStyle>
  <tcBdr>
    …
  </tcBdr>
  <fill>
    …
  </fill>
</tcStyle>
```

In this example, we see that a set of borders for the cell along with a cell fill are being defined. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleCellStyle) is located in §A.4.1. *end note*]

20.1.4.2.30 tcTxStyle (Table Cell Text Style)

This element defines the text properties associated with the text contained within a table cell.

[*Example*: Consider the following example of a table cell text style in use within DrawingML:

```
<tcTxStyle b="on">
  <fontRef idx="minor">
    <scrgbClr r="0" g="0" b="0"/>
  </fontRef>
  <schemeClr val="lt1"/>
</tcTxStyle>
```

In this example, we define the text within the cell to be bold and reference the themed minor font and to also be the light 1 color. *end example*]

Attributes	Description
b (Bold)	Specifies if the text is to be bolded. The possible values for this attribute are defined by the ST_OnOffStyleType simple type (§20.1.10.36).
i (Italic)	Specifies if the text is to be italicized. The possible values for this attribute are defined by the ST_OnOffStyleType simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(§20.1.10.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyleTextStyle) is located in §A.4.1. *end note*]

20.1.4.2.31 tl2br (Top Left to Bottom Right Border)

This element defines the line properties associated with the border which goes from the top-left to the bottom-right corner in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<tl2br>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</tl2br>
```

In this example, we see the border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.32 top (Top Border)

This element defines the line properties associated with the top border in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<top>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</top>
```

In this example, we see the top border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.4.2.33 tr2bl (Top Right to Bottom Left Border)

This element defines the line properties associated with the border which goes from the top-right to the bottom-left corner in a table cell.

[*Example*: Consider the following example of the top border in use within DrawingML:

```
<tr2bl>
  <ln w="12700" cmpd="sng">
    <solidFill>
      <schemeClr val="accent1"/>
    </solidFill>
  </ln>
</tr2bl>
```

In this example, we see the border on a table cell to be a single 1pt line which is colored accent 1. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ThemeableLineStyle) is located in §A.4.1. *end note*]

20.1.4.2.34 wholeTbl (Whole Table)

This element contains formatting options which are applied to the table as a whole when it is in its default state with no formatting options (first row, last row, etc) enabled.

[*Example*: Consider the following example of whole table being used within DrawingML:

```
<wholeTbl>
  <tcTxStyle>
    …
  </tcTxStyle>
  <tcStyle>
    …
  </tcStyle>
</wholeTbl>
```

In this example, we see definitions for the text and the cells within the table. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TablePartStyle) is located in §A.4.1. *end note*]

20.1.5 3D

The 3D portion of the DrawingML framework allows for the describing of a 3D scene to be placed within a document. This 3D scene can be described using text and shape objects along with various lighting, material and camera settings.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.5.1 anchor (Anchor Point)

This element specifies a point in 3D space. This point is the point in space that anchors the backdrop plane. Please see the example in the backdrop (§20.1.5.2) definition for an in depth explanation of this element.

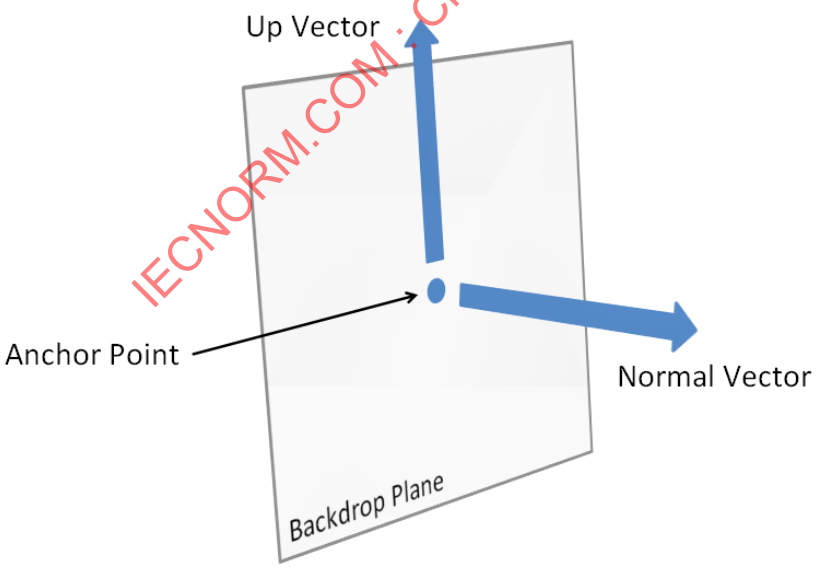
Attributes	Description
x (X-Coordinate in 3D)	X-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Coordinate in 3D)	Y-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
z (Z-Coordinate in 3D)	Z-Coordinate in 3D space. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point3D) is located in §A.4.1. *end note*]

20.1.5.2 backdrop (Backdrop Plane)

This element defines a plane in which effects, such as glow and shadow, are applied in relation to the shape they are being applied to. The points and vectors contained within the backdrop define a plane in 3D space.

[*Example*: Consider the following image as an explanation of the backdrop plane definition:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

In this image we see a plane being defined by an anchor point, the vector normal to the face of the plane and a vector pointing up in relation to the plane. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Backdrop) is located in §A.4.1. *end note*]

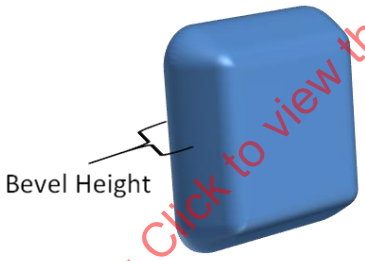
20.1.5.3 bevelB (Bottom Bevel)

This element holds the properties associated with defining a bevel on the bottom or back face of a shape.

[*Example*: Consider the following example of an sp3d containing a bottom bevel.

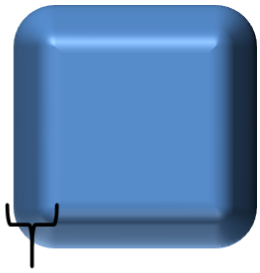
```
<a:sp3d>
  <a:bevelB w="139700" h="127000" prst="coolSlant"/>
</a:sp3d>
```

In this example, we see a bottom bevel being defined with a preset bevel type along with a custom width and height. *end example*]

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the height of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 Bevel Width In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

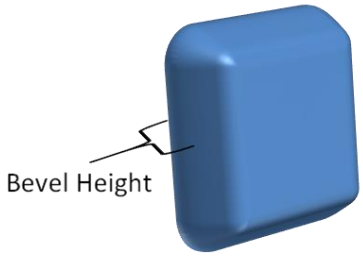
20.1.5.4 bevelT (Top Bevel)

This element holds the properties associated with defining a bevel on the top or front face of a shape.

[*Example*: Consider the following example of an sp3d containing a top bevel.

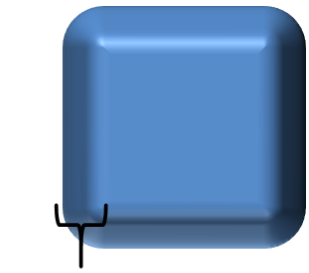
```
<a:sp3d>
  <a:bevelT w="139700" h="127000" prst="coolSlant"/>
</a:sp3d>
```

In this example, we see a top bevel being defined with a preset bevel type along with a custom width and height. *end example*]

Attributes	Description
h (Height)	Specifies the height of the bevel, or how far above the shape it is applied. [*Example*: Consider the following example bevel  Bevel Height In this example, we see the height of an example bevel on a shape. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Bevel)	Specifies the preset bevel type which defines the look of the bevel. The possible values for this attribute are defined by the ST_BevelPresetType simple type (§20.1.10.9).
w (Width)	Specifies the width of the bevel, or how far into the shape it is applied. [*Example*: Consider the following example bevel  In this example, we see the width of an example bevel on a shape. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bevel) is located in §A.4.1. *end note*]

20.1.5.5 camera (Camera)

This element defines the placement and properties of the camera in the 3D scene. The camera position and properties modify the view of the scene.

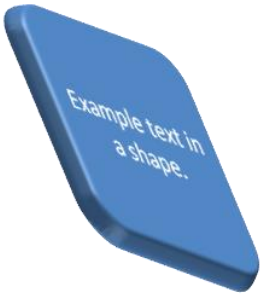
[*Example*: Consider the following example of a camera in DrawingML:

```
<a:camera prst="orthographicFront">
  <a:rot lat="19902513" lon="17826689" rev="1362739"/>
</a:camera>
```

In this example, we see a preset camera being defined along with a rotation containing latitude, longitude, and revolution overrides provided that further rotate the camera around the scene. The effect of this camera can be seen on the following shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



end example]

Attributes	Description
fov (Field of View)	Provides an override for the default field of view for the camera. Different perspectives can be obtained by modifying this attribute. [*Example*: Consider the following example of a fov in DrawingML: `<a:camera prst="perspectiveContrastingRightFacing" fov="6900000"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> </a:camera>` In this example, we see a fov being defined which modifies the default fov for the preset camera. *end example*] The possible values for this attribute are defined by the ST_FOVAngle simple type (§20.1.10.26).
prst (Preset Camera Type)	Defines the preset camera that is being used by the camera element. The preset camera defines a starting point for common preset rotations in space. [*Example*: Consider the following example of a prst in DrawingML: `<a:camera prst="perspectiveContrastingRightFacing" fov="6900000"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> </a:camera>` In this example, we see a prst being defined as `perspectiveContrastingRightFacing`. *end example*] The possible values for this attribute are defined by the ST_PresetCameraType simple type (§20.1.10.47).
zoom (Zoom)	Defines the zoom factor of a given camera element. The zoom modifies the scene as a whole and zooms in or out accordingly. [*Example*: Consider the following example of a zoom in DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:camera prst="perspectiveContrastingRightFacing" fov="6900000" zoom="200%"> <a:rot lat="1200000" lon="18000000" rev="1200000"/> /a:camera> In this example, we see a zoom being used which zooms the scene by 200%. *end example*] The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Camera) is located in §A.4.1. *end note*]

20.1.5.6 contourClr (Contour Color)

This element defines the color for the contour on a shape. The contour of a shape is a solid filled line which surrounds the outer edges of the shape.

[*Example*: Consider the following example of a contour defined on a shape which includes a contourClr. Lighting characteristics applied to the shape are ignored when it comes to the contour on the shape.

```
<a:sp3d contourW="101600" prstMaterial="plastic">
  <a:bevelT w="254000" h="254000"/>
  <a:bevelB w="254000" h="254000"/>
  <a:contourClr>
    <a:schemeClr val="bg1"/>
  </a:contourClr>
</a:sp3d>
```

In this example, we see a contour defined on a shape with a top and bottom bevel defined. In the image below, the contour is the white ring around the shape.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.5.7 extrusionClr (Extrusion Color)

This element defines the color of the extrusion applied to a shape. The extrusion on a shape is an artificial height applied to the geometry.

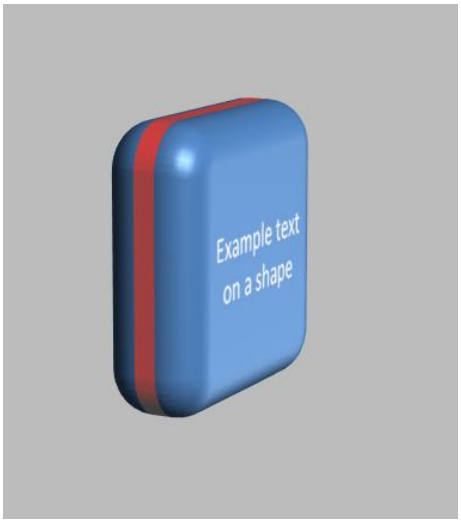
[*Example*: Consider the following example of an extrusion which takes advantage of the extrusionClr. Lighting characteristics that are applied to the shape are also applied to the extrusion on the shape.

```
<a:sp3d extrusionH="139700" prstMaterial="plastic">
  <a:bevelT w="254000" h="254000"/>
  <a:bevelB w="254000" h="254000"/>
  <a:extrusionClr>
  <a:srgbClr val="FF0000"/>
  </a:extrusionClr>
</a:sp3d>
```

In this example, we see the extrusion color defined as red which can also be shown applied to the shape in the following image:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.5.8 flatTx (No text in 3D scene)

Keep text out of 3D scene entirely.

Attributes	Description
z (Z Coordinate)	Specifies the Z coordinate to be used in positioning the flat text within the 3D scene. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FlatText) is located in §A.4.1. *end note*]

20.1.5.9 lightRig (Light Rig)

This element defines the light rig associated with the table. The light rig comes into play when there is a 3D bevel applied to a cell. When 3D is used, the light rig defines the lighting properties associated with the scene.

Attributes	Description
dir (Direction)	Defines the direction from which the light rig is oriented in relation to the scene. [*Example*: Consider the following example of dir being used in a light rig: `<a:lightRig rig="threePt" dir="t"/>` In this example, we define the direction to be `top`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

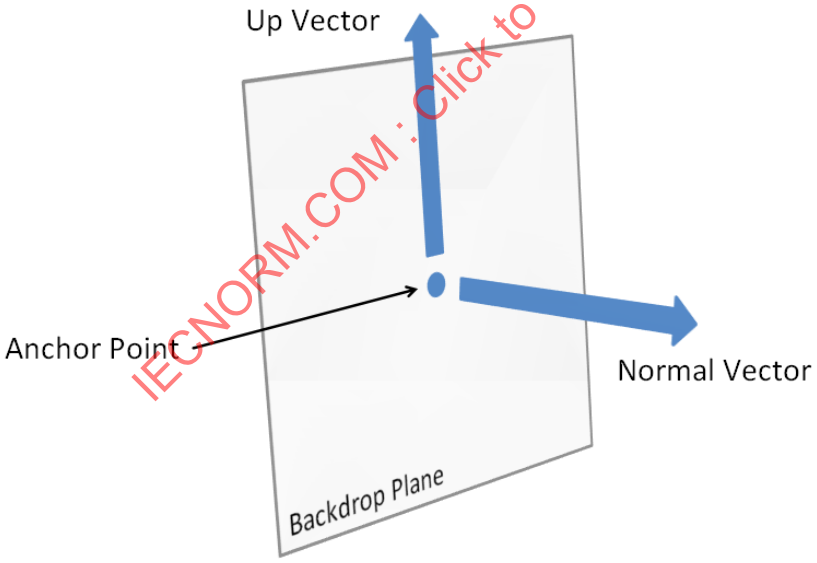
Attributes	Description
	The possible values for this attribute are defined by the ST_LightRigDirection simple type (§20.1.10.29).
rig (Rig Preset)	Defines the preset type of light rig which is to be applied to the scene. [*Example*: Consider the following example of rig being used in a light rig: `<a:lightRig rig="threePt" dir="t"/>` In this example, we define the rig to be a `threePt` rig. *end example*] The possible values for this attribute are defined by the ST_LightRigType simple type (§20.1.10.30).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LightRig) is located in §A.4.1. *end note*]

20.1.5.10 norm (Normal)

This element defines a normal vector. To be more precise, this attribute defines a vector normal to the face of the backdrop plane.

[*Example*: Consider the following image as an example of what a normal vector is in relation to the backdrop plane:



end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
dx (Distance along X-axis in 3D)	Distance along X-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dy (Distance along Y-axis in 3D)	Distance along Y-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dz (Distance along Z-axis in 3D)	Distance along Z-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector3D) is located in §A.4.1. *end note*]

20.1.5.11 rot (Rotation)

This element defines a rotation in 3D space. A rotation in DrawingML is defined through the use of a latitude coordinate, a longitude coordinate, and a revolution about the axis as the latitude and longitude coordinates.

[*Example*: Consider the following example of a rotation defined by the rot elements being used in a lightRig in DrawingML:

```
<a:lightRig rig="twoPt" dir="t">
  <a:rot lat="0" lon="0" rev="6000000"/>
</a:lightRig>
```

In this example, we have only a revolution applied to the light rig which rotates it around it's center axis. *end example*]

Attributes	Description
lat (Latitude)	Defines the latitude value of the rotation. [*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the lat to be equal to `0`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
lon (Longitude)	Defines the longitude value of the rotation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	[*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the lon to be equal to `0`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
rev (Revolution)	This attributes defines the revolution around the central axis in the rotation. [*Example*: Consider the following example of a rot in DrawingML: `<a:rot lat="0" lon="0" rev="6000000"/>` In this example, we set the rev to be equal to `6000000`. *end example*] The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SphereCoords) is located in §A.4.1. *end note*]

20.1.5.12 sp3d (Apply 3D shape properties)

This element defines the 3D properties associated with a particular shape in DrawingML. The 3D properties which can be applied to a shape are top and bottom bevels, a contour and an extrusion.

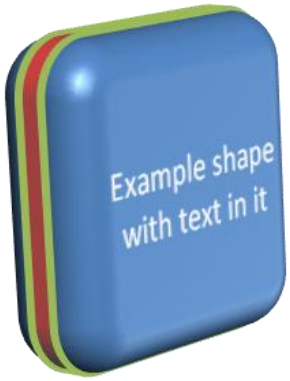
[*Example*: Consider the following example of an sp3d in DrawingML:

```
<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic">
  <a:bevelT w="254000" h="254000"/>
  <a:bevelB w="254000" h="254000"/>
  <a:extrusionClr>
    <a:srgbClr val="FF0000"/>
  </a:extrusionClr>
  <a:contourClr>
    <a:schemeClr val="accent3"/>
  </a:contourClr>
</a:sp3d>
```

In this example, we see an sp3d defined which contains information defining both a top and bottom bevel, along with an extrusion and contour on the shape. The following image illustrates a shape with the applied sp3d:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved



end example]

Attributes	Description
contourW (Contour Width)	Defines the width of the contour on the shape. [*Example*: Consider the following example of a contourW in use within the sp3d element: `<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"> <a:bevelT w="254000" h="254000"/> <a:bevelB w="254000" h="254000"/> <a:extrusionClr> <a:srgbClr val="FF0000"/> </a:extrusionClr> <a:contourClr> <a:schemeClr val="accent3"/> </a:contourClr> </a:sp3d>` In this example, we see a countourW defined as `50800`. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
extrusionH (Extrusion Height)	Defines the height of the extrusion applied to the shape. [*Example*: Consider the following example of an extrusionH in use within the sp3d element: `<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"> <a:bevelT w="254000" h="254000"/> <a:bevelB w="254000" h="254000"/> <a:extrusionClr> <a:srgbClr val="FF0000"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:extrusionClr> <a:contourClr> <a:schemeClr val="accent3"/> </a:contourClr> </a:sp3d> In this example, we see a extrusionH defined as 165100. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prstMaterial (Preset Material Type)	Defines the preset material which is combined with the lighting properties to give the final look and feel of a shape. [*Example*: Consider the following example of a prstMaterial in use within the sp3d element: <a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"> <a:bevelT w="254000" h="254000"/> <a:bevelB w="254000" h="254000"/> <a:extrusionClr> <a:srgbClr val="FF0000"/> </a:extrusionClr> <a:contourClr> <a:schemeClr val="accent3"/> </a:contourClr> </a:sp3d> In this example, we see a prstMaterial defined as plastic. *end example*] The possible values for this attribute are defined by the ST_PresetMaterialType simple type (§20.1.10.50).
z (Shape Depth)	Defines the z coordinate for the 3D shape. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

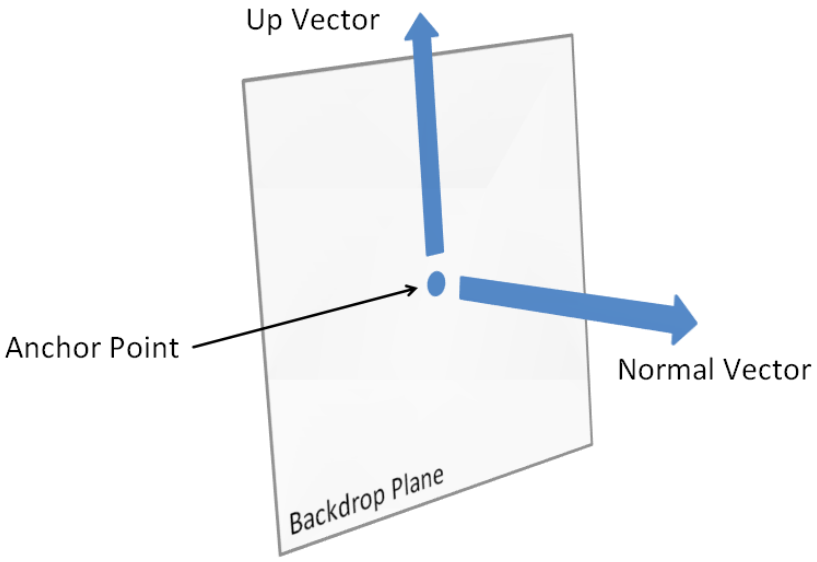
[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape3D) is located in §A.4.1. *end note*]

20.1.5.13 up (Up Vector)

This element defines a vector representing up. To be more precise, this attribute defines a vector representing up in relation to the face of the backdrop plane.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following image as an example of what an up vector is in relation to the backdrop plane:



end example]

Attributes	Description
dx (Distance along X-axis in 3D)	Distance along X-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dy (Distance along Y-axis in 3D)	Distance along Y-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
dz (Distance along Z-axis in 3D)	Distance along Z-axis in 3D The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector3D) is located in §A.4.1. *end note*]

20.1.6 Shared Style Sheet

The shared style sheet aspects contained within DrawingML are responsible for containing formatting options and styles which can be used by applications to define a certain look or feel to documents. The shared style sheet can be used by any document category ([*Note*: For example, a presentation. *end note*]) to pull visual information from which formats the document in a certain way, or theme. The shared style sheet contains information that is not document-category specific.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.6.1 clrMap (Color Map)

This element specifics the color mapping layer which allows a user to define colors for background and text. This allows for swapping out of light/dark colors for backgrounds and the text on top of the background in order to maintain readability of the text On a deeper level, this specifies exactly which colors the first 12 values refer to in the color scheme.

[*Example*: Consider the following example of a color map in use:

```
<clrMap bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
  accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
  accent6="accent6" hlink="hlink" folHlink="folHlink"/>
```

In this example, we see that bg1 is mapped to lt1, tx1 is mapped to dk1, and so on. *end example*]

Attributes	Description
accent1 (Accent 1)	Specifies a color defined which is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent2 (Accent 2)	Specifies a color defined which is associated as the accent 2 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent3 (Accent 3)	Specifies a color defined which is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent4 (Accent 4)	Specifies a color defined which is associated as the accent 4 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent5 (Accent 5)	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent6 (Accent 6)	Specifies a color defined which is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg1 (Background 1)	A color defined which is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg2 (Background 2)	Specifies a color defined which is associated as the second background color.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
folHlink (Followed Hyperlink)	Specifies a color defined which is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
hlink (Hyperlink)	Specifies a color defined which is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx1 (Text 1)	Specifies a color defined which is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx2 (Text 2)	Specifies a color defined which is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

20.1.6.2 clrScheme (Color Scheme)

This element defines a set of colors which are referred to as a color scheme. The color scheme is responsible for defining a list of twelve colors. The twelve colors consist of six accent colors, two dark colors, two light colors and a color for each of a hyperlink and followed hyperlink.

The Color Scheme Color elements appear in a sequence. The following listing shows the index value and corresponding Color Name.

Sequence Index	Element (Color) Name
0	dk1 (Dark 1)
1	lt1 (Light 1)
2	dk2 (Dark 2)
3	lt2 (Light 2)
4	accent1 (Accent 1)
5	accent2 (Accent 2)
6	accent3 (Accent 3)
7	accent4 (Accent 4)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Sequence Index	Element (Color) Name
8	accent5 (Accent 5)
9	accent6 (Accent 6)
10	hlink (Hyperlink)
11	folHlink (Followed Hyperlink)

[*Example*: Consider the following example of a color scheme defined in DrawingML:

```
<clrScheme name="sample">
  <dk1>
    <sysClr val="windowText"/>
  </dk1>
  <lt1>
    <sysClr val="window"/>
  </lt1>
  <dk2>
    <srgbClr val="04617B"/>
  </dk2>
  <lt2>
    <srgbClr val="DBF5F9"/>
  </lt2>
  <accent1>
    <srgbClr val="0F6FC6"/>
  </accent1>
  <accent2>
    <srgbClr val="009DD9"/>
  </accent2>
  <accent3>
    <srgbClr val="0BD0D9"/>
  </accent3>
  <accent4>
    <srgbClr val="10CF9B"/>
  </accent4>
  <accent5>
    <srgbClr val="7CCA62"/>
  </accent5>
  <accent6>
    <srgbClr val="A5C249"/>
  </accent6>
  <hlink>
    <srgbClr val="FF9800"/>
  </hlink>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <folHlink>
      <srgbClr val="F45511"/>
    </folHlink>
  </clrScheme>
```

In this example, are defined the 12 theme colors in the sample color scheme. *end example*]

Attributes	Description
name (Name)	The common name for this color scheme. This name can show up in the user interface in a list of color schemes. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorScheme) is located in §A.4.1. *end note*]

20.1.6.3 custClrLst (Custom Color List)

This element allows for a custom color palette to be created and which shows up alongside other color schemes. This can be very useful, for example, when someone would like to maintain a corporate color palette.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomColorList) is located in §A.4.1. *end note*]

20.1.6.4 extraClrScheme (Extra Color Scheme)

This element defines an auxiliary color scheme, which includes both a color scheme and color mapping. This is mainly used for backward compatibility concerns and roundtrips information required by earlier versions.

[*Example*: Consider the following example of an extra color scheme in use in DrawingML:

```
<extraClrScheme>
  <clrScheme name="extraColorSchemeSample">
    <dk1>
      <sysClr val="windowText"/>
    </dk1>
    <lt1>
      <sysClr val="window"/>
    </lt1>
    <dk2>
      <srgbClr val="04617B"/>
    </dk2>
    <lt2>
      <srgbClr val="DBF5F9"/>
    </lt2>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <accent1>
        <srgbClr val="0F6FC6"/>
      </accent1>
      <accent2>
        <srgbClr val="009DD9"/>
      </accent2>
      <accent3>
        <srgbClr val="0BD0D9"/>
      </accent3>
      <accent4>
        <srgbClr val="10CF9B"/>
      </accent4>
      <accent5>
        <srgbClr val="7CCA62"/>
      </accent5>
      <accent6>
        <srgbClr val="A5C249"/>
      </accent6>
      <hlink>
        <srgbClr val="FF9800"/>
      </hlink>
      <folHlink>
        <srgbClr val="F45511"/>
      </folHlink>
    </clrScheme>
    <clrMap bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
    accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
    accent6="accent6" hlink="hlink" folHlink="folHlink"/>
  </extraClrScheme>
```

In this example, the extra color scheme contains a color scheme and a color map for that color scheme. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorSchemeAndMapping) is located in §A.4.1. *end note*]

20.1.6.5 extraClrSchemeLst (Extra Color Scheme List)

This element is a container for the list of extra color schemes present in a document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorSchemeList) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.6.6 masterClrMapping (Master Color Mapping)

This element is a part of a choice for which color mapping is used within the document. There is also defined an overrideClrMapping (§20.1.6.8) element which, when specified, the override is used rather than the color mapping defined in the master. If this element is specified, then we specifically use the color mapping defined in the master.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmptyElement) is located in §A.4.1. *end note*]

20.1.6.7 objectDefaults (Object Defaults)

This element allows for the definition of default shape, line, and textbox formatting properties. An application can use this information to format a shape (or text) initially on insertion into a document.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ObjectStyleDefaults) is located in §A.4.1. *end note*]

20.1.6.8 overrideClrMapping (Override Color Mapping)

This element provides an override for the color mapping in a document. When defined, this color mapping is used in place of the already defined color mapping, or master color mapping. This color mapping is defined in the same manner as the other mappings within this document.

[*Example*: Consider the following example of an override color mapping in DrawingML:

```
<overrideClrMapping bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" accent1="accent1"
  accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5"
  accent6="accent6" hlink="hlink" folHlink="folHlink"/>
```

end example]

Attributes	Description
accent1 (Accent 1)	Specifies a color defined which is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent2 (Accent 2)	Specifies a color defined which is associated as the accent 2 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent3 (Accent 3)	Specifies a color defined which is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent4 (Accent 4)	Specifies a color defined which is associated as the accent 4 color.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent5 (Accent 5)	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent6 (Accent 6)	Specifies a color defined which is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg1 (Background 1)	A color defined which is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg2 (Background 2)	Specifies a color defined which is associated as the second background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
folHlink (Followed Hyperlink)	Specifies a color defined which is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
hlink (Hyperlink)	Specifies a color defined which is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx1 (Text 1)	Specifies a color defined which is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx2 (Text 2)	Specifies a color defined which is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

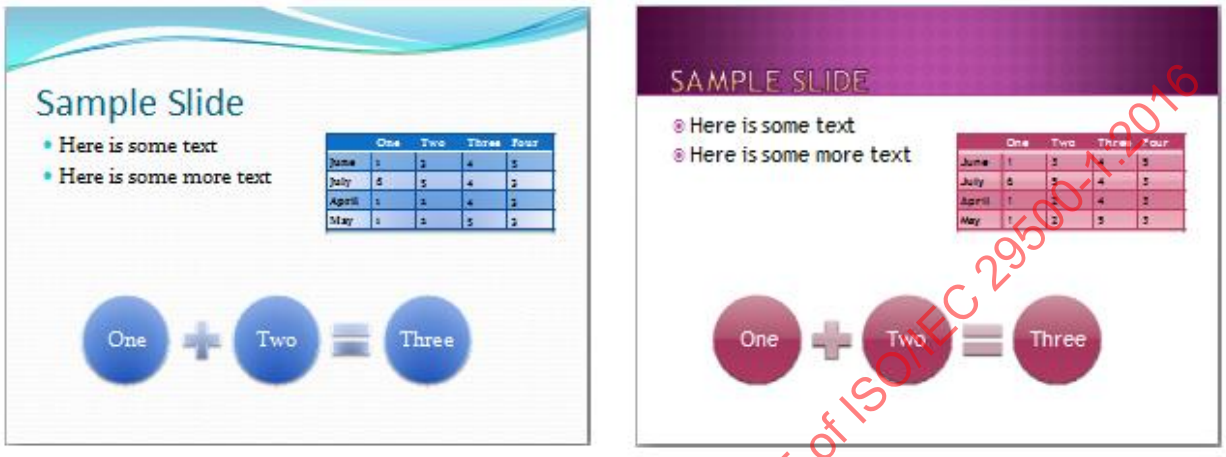
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.6.9 theme (Theme)

This element defines the root level complex type associated with a shared style sheet (or theme). This element holds all the different formatting options available to a document through a theme and defines the overall look and feel of the document when themed objects are used within the document.

[*Example*: Consider the following image as an example of different themes in use applied to a presentation:



In this example, we see how a theme can affect font, colors, backgrounds, fills, and effects for different objects in a presentation. *end example*]

Attributes	Description
name (Name)	Specifies the name given to the theme. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeStyleSheet) is located in §A.4.1. *end note*]

20.1.6.10 themeElements (Theme Elements)

This element defines the theme formatting options for the theme and is the workhorse of the theme. This is where the bulk of the shared theme information is contained and used by a document. This element contains the color scheme, font scheme, and format scheme elements which define the different formatting aspects of what a theme defines.

[*Example*: Consider the following example of a theme elements defined in DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<themeElements>
  <clrScheme name="sample">
…
  </clrScheme>
  <fontScheme name="sample">
…
  </fontScheme>
  <fmtScheme name="sample">
    <fillStyleLst>
…
    </fillStyleLst>
    <lnStyleLst>
…
    </lnStyleLst>
    <effectStyleLst>
…
    </effectStyleLst>
    <bgFillStyleLst>
…
    </bgFillStyleLst>
  </fmtScheme>
</themeElements>
```

In this example, we see the basic structure of how a theme elements is defined and have left out the true guts of each individual piece to save room. Each part (color scheme, font scheme, format scheme) is defined elsewhere within DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BaseStyles) is located in §A.4.1. *end note*]

20.1.6.11 themeManager (Theme Manager)

The starting part for a theme file.

[*Note*: The W3C XML Schema definition of this element's content model (CT_EmptyElement) is located in §A.4.1. *end note*]

20.1.6.12 themeOverride (Theme Override)

This element allows for an override which changes just the colors, fonts, or effects of a single object, like a table for example. Currently it is used only to control overrides on the non-top-level masters within a presentation.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BaseStylesOverride) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.7 Coordinate Systems and Transformations

The following elements are used to reflect dimensions, scaling, location, rotation, and flip information on groups and individual shapes respectively.

20.1.7.1 chExt (Child Extents)

This element specifies the size dimensions of the child extents rectangle and is used for calculations of grouping, scaling, and rotation behavior of shapes placed within a group.

Attributes	Description
cx (Extent Length)	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width)	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.1.7.2 chOff (Child Offset)

This element specifies the location of the child extents rectangle and is used for calculations of grouping, scaling, and rotation behavior of shapes placed within a group.

Attributes	Description
x (X-Axis	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Coordinate)	by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate)	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.1.7.3 ext (Extents)

This element specifies the size of the bounding box enclosing the referenced object.

Attributes	Description
cx (Extent Length)	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of `1828800` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
cy (Extent Width)	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of `200000` EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.1.7.4 off (Offset)

This element specifies the location of the bounding box of an object. Effects on an object are not included in this bounding box.

Attributes	Description
x (X-Axis Coordinate)	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate)	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.1.7.5 xfrm (2D Transform for Grouped Objects)

This element is nearly identical to the representation of 2-D transforms for ordinary shapes (§20.1.7.6). The only addition is a member to represent the Child offset and the Child extents.

Attributes	Description
flipH (Horizontal Flip)	Horizontal flip. When true, this attribute defines that the group is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip)	Vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation)	Rotation. Specifies the clockwise rotation of a group in 1/64000 of a degree. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupTransform2D) is located in §A.4.1. *end note*]

20.1.7.6 xfrm (2D Transform for Individual Objects)

This element represents 2-D transforms for ordinary shapes.

Attributes	Description
flipH (Horizontal Flip)	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip)	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation)	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.8 Shape Fills, Effects, and Line Properties

This portion of the DrawingML framework describes effects defining the visual appearance of shapes and lines. Shapes can be filled in a variety of ways, with images, solid colors, gradients, or pattern fills. In addition, several visual effects can alter the appearance of a shape, and multiple effects can be combined together. Lines also can have special properties defining how they are rendered, included a dashed appearance or decorations at the line ends. This section documents the elements that define these properties and effects for shapes and lines.

20.1.8.1 alphaBiLevel (Alpha Bi-Level Effect)

This element represents an Alpha Bi-Level Effect.

Alpha (Opacity) values less than the threshold are changed to 0 (fully transparent) and alpha values greater than or equal to the threshold are changed to 100% (fully opaque).

Attributes	Description
thresh (Threshold)	Specifies the threshold value for the alpha bi-level effect. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaBiLevelEffect) is located in §A.4.1. *end note*]

20.1.8.2 alphaCeiling (Alpha Ceiling Effect)

This element represents an alpha ceiling effect.

Alpha (opacity) values greater than zero are changed to 100%. In other words, anything partially opaque becomes fully opaque.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaCeilingEffect) is located in §A.4.1. *end note*]

20.1.8.3 alphaFloor (Alpha Floor Effect)

This element represents an alpha floor effect.

Alpha (opacity) values less than 100% are changed to zero. In other words, anything partially transparent becomes fully transparent.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaFloorEffect) is located in §A.4.1. *end note*]

20.1.8.4 alphaInv (Alpha Inverse Effect)

This element represents an alpha inverse effect.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Alpha (opacity) values are inverted by subtracting from 100%.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaInverseEffect) is located in §A.4.1. *end note*]

20.1.8.5 alphaMod (Alpha Modulate Effect)

This element represents an alpha modulate effect.

Effect alpha (opacity) values are multiplied by a fixed percentage. The effect container specifies an effect containing alpha values to modulate.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaModulateEffect) is located in §A.4.1. *end note*]

20.1.8.6 alphaModFix (Alpha Modulate Fixed Effect)

This element represents an alpha modulate fixed effect.

Effect alpha (opacity) values are multiplied by a fixed percentage.

Attributes	Description
amt (Amount)	Specifies the percentage amount to scale the alpha. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaModulateFixedEffect) is located in §A.4.1. *end note*]

20.1.8.7 alphaOutset (Alpha Inset/Outset Effect)

This element specifies an alpha outset/inset effect.

This is equivalent to an alpha ceiling, followed by alpha blur, followed by either an alpha ceiling (positive radius) or alpha floor (negative radius).

Attributes	Description
rad (Radius)	Specifies the radius of outset/inset. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaOutsetEffect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.8.8 alphaRepl (Alpha Replace Effect)

This element specifies an alpha replace effect.

Effect alpha (opacity) values are replaced by a fixed alpha.

Attributes	Description
a (Alpha)	Specifies the new opacity value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AlphaReplaceEffect) is located in §A.4.1. *end note*]

20.1.8.9 bevel (Line Join Bevel)

This element specifies a Bevel Line Join.

A bevel joint specifies that an angle joint is used to connect lines.

[*Example*:

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinBevel) is located in §A.4.1. *end note*]

20.1.8.10 bgClr (Background color)

This element specifies the background color of a Pattern fill.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.11 biLevel (Bi-Level (Black/White) Effect)

This element specifies a bi-level (black/white) effect. Input colors whose luminance is less than the specified threshold value are changed to black. Input colors whose luminance are greater than or equal the specified value are set to white. The alpha effect values are unaffected by this effect.

Attributes	Description
thresh (Threshold)	Specifies the luminance threshold for the Bi-Level effect. Values greater than or equal to the threshold are set to white. Values lesser than the threshold are set to black.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BiLevelEffect) is located in §A.4.1. *end note*]

20.1.8.12 blend (Blend Effect)

This element specifies a blend of several effects. The container specifies the raw effects to blend while the blend mode specifies how the effects are to be blended.

Attributes	Description
blend (Blend Mode)	Specifies how to blend the two effects. The possible values for this attribute are defined by the ST_BlendMode simple type (§20.1.10.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlendEffect) is located in §A.4.1. *end note*]

20.1.8.13 blip (Blip)

This element specifies the existence of an image (binary large image or picture) and contains a reference to the image data.

Attributes	Description
cstate (Compression State)	Specifies the compression state with which the picture is stored. This allows the application to specify the amount of compression that has been applied to a picture. The possible values for this attribute are defined by the ST_BlipCompression simple type (§20.1.10.12).
embed (Embedded Picture Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the identification information for an embedded picture. This attribute is used to specify an image that resides locally within the file. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
link (Linked Picture Reference)	Specifies the identification information for a linked picture. This attribute is used to specify an image that does not reside within this file.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Blip) is located in §A.4.1. *end note*]

20.1.8.14 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<p:pic>
  …
  <p:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </p:blipFill>
  …
</p:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<p:pic>
  …
  <p:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </p:blipFill>
  <p:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </p:spPr>
  …
</p:pic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting)	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape)	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

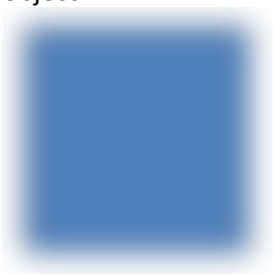
20.1.8.15 blur (Blur Effect)

This element specifies a blur effect that is applied to the entire shape, including its fill. All color channels, including alpha, are affected.

Attributes	Description
grow (Grow Bounds)	Specifies whether the bounds of the object should be grown as a result of the blurring. True indicates the bounds are grown while false indicates that they are not. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	With grow set to false, the blur effect does not extend beyond the original bounds of the object: With grow set to true, the blur effect can extend beyond the original bounds of the object:  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rad (Radius)	Specifies the radius of blur. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlurEffect) is located in §A.4.1. *end note*]

20.1.8.16 clrChange (Color Change Effect)

This element specifies a Color Change Effect. Instances of clrFrom are replaced with instances of clrTo.

Attributes	Description
useA (Consider Alpha Values)	Specifies whether alpha values are considered for the effect. Effect alpha values are considered if useA is true, else they are ignored. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorChangeEffect) is located in §A.4.1. *end note*]

20.1.8.17 clrFrom (Change Color From)

This element specifies a color getting removed in a color change effect. It is the "from" or source input color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.18 clrRepl (Solid Color Replacement)

This element specifies a solid color replacement value. All effect colors are changed to a fixed color. Alpha values are unaffected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorReplaceEffect) is located in §A.4.1. *end note*]

20.1.8.19 clrTo (Change Color To)

This element specifies the color which replaces the clrFrom in a clrChange effect. This is the "target" or "to" color in the color change effect.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.20 cont (Effect Container)

This element specifies an Effect Container. It is a list of effects.

Attributes	Description
name (Name)	Specifies an optional name for this list of effects, so that it can be referred to later. Shall be unique across all effect trees and effect containers. The possible values for this attribute are defined by the W3C XML Schema token datatype.
type (Effect Container Type)	Specifies the kind of container, either sibling or tree. The possible values for this attribute are defined by the ST_EffectContainerType simple type (§20.1.10.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectContainer) is located in §A.4.1. *end note*]

20.1.8.21 custDash (Custom Dash)

This element specifies a custom dashing scheme. It is a list of dash stop elements which represent building block atoms upon which the custom dashing scheme is built.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DashStopList) is located in §A.4.1. *end note*]

20.1.8.22 ds (Dash Stop)

This element specifies a dash stop primitive. Dashing schemes are built by specifying an ordered list of dash stop primitive. A dash stop primitive consists of a dash and a space.

Attributes	Description
d (Dash Length)	Specifies the length of the dash relative to the line width. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).
sp (Space Length)	Specifies the length of the space relative to the line width. The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DashStop) is located in §A.4.1. *end note*]

20.1.8.23 duotone (Duotone Effect)

This element specifies a duotone effect.

For each pixel, combines clr1 and clr2 through a linear interpolation to determine the new color for that pixel.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DuotoneEffect) is located in §A.4.1. *end note*]

20.1.8.24 effect (Effect)

This element specifies a reference to an existing effect container.

Attributes	Description
ref (Reference)	Specifies the reference. Its value can be the name of an effect container, or one of four special references: fill - refers to the fill effect line - refers to the line effect fillLine - refers to the combined fill and line effects children - refers to the combined effects from logical child shapes or text The possible values for this attribute are defined by the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectReference) is located in §A.4.1. *end note*]

20.1.8.25 effectDag (Effect Container)

This element specifies a list of effects. Effects are applied in the order specified by the container type (sibling or tree).

[*Note:* An effectDag element can contain multiple effect containers as child elements. Effect containers with different styles can be combined in an effectDag to define a directed acyclic graph (DAG) that specifies the order in which all effects are applied. *end note*]

Attributes	Description
name (Name)	Specifies an optional name for this list of effects, so that it can be referred to later. Shall be unique across all effect trees and effect containers. The possible values for this attribute are defined by the W3C XML Schema token datatype.
type (Effect Container Type)	Specifies the kind of container, either sibling or tree. The possible values for this attribute are defined by the ST_EffectContainerType simple type (§20.1.10.22).

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectContainer) is located in §A.4.1. *end note*]

20.1.8.26 effectLst (Effect Container)

This element specifies a list of effects. Effects in an effectLst are applied in the default order by the rendering engine. The following diagrams illustrate the order in which effects are applied, both for shapes and for group shapes.

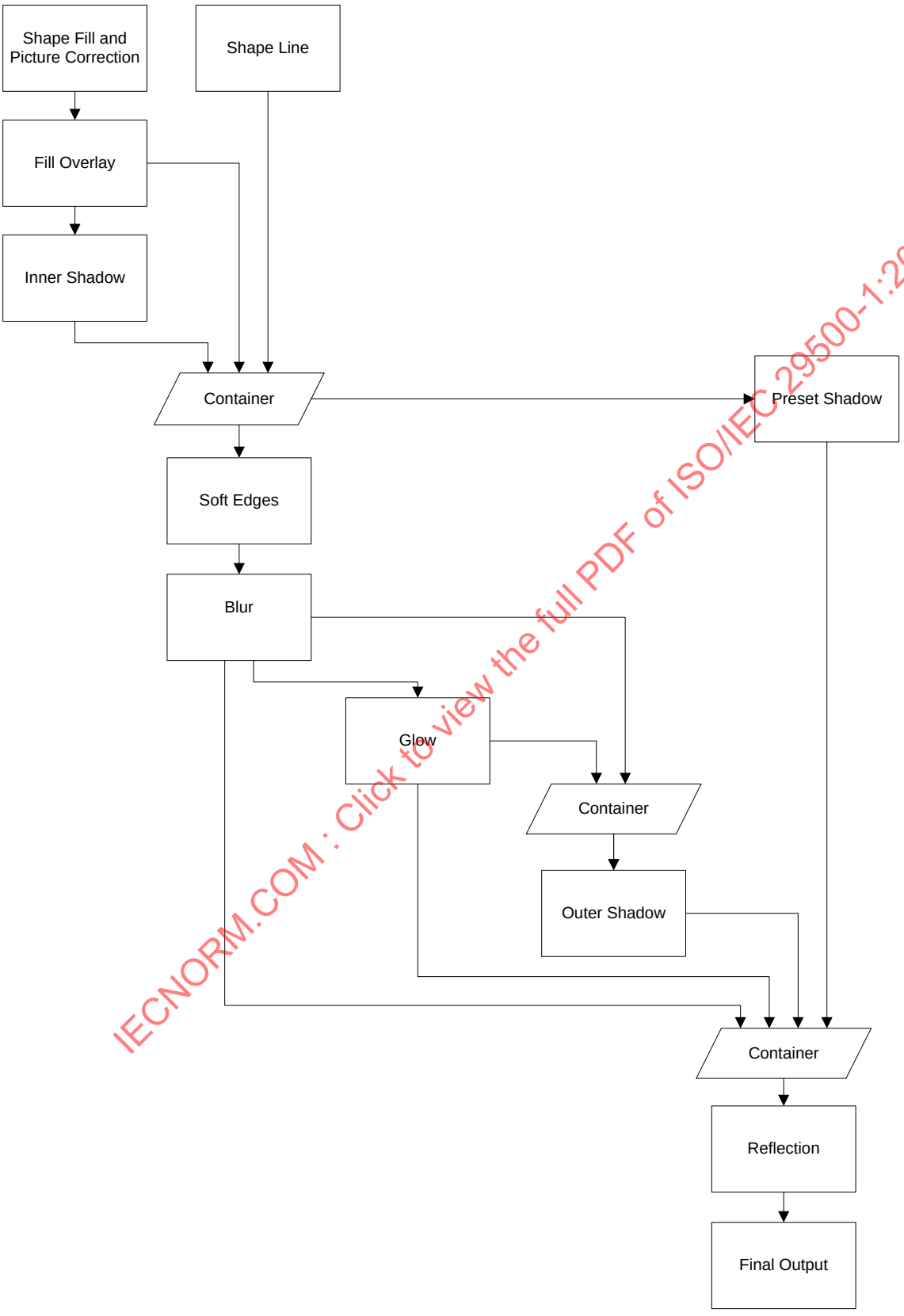
[*Note:* The output of many effects does not include the input shape. For effects that should be applied to the result of previous effects as well as the original shape, a container is used to group the inputs together. *end note*]

[*Example:* Outer Shadow is applied both to the original shape and the original shape's glow. The result of blur contains the original shape, while the result of glow contains only the added glow. Therefore, a container that groups the blur result with the glow result is used as the input to Outer Shadow. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

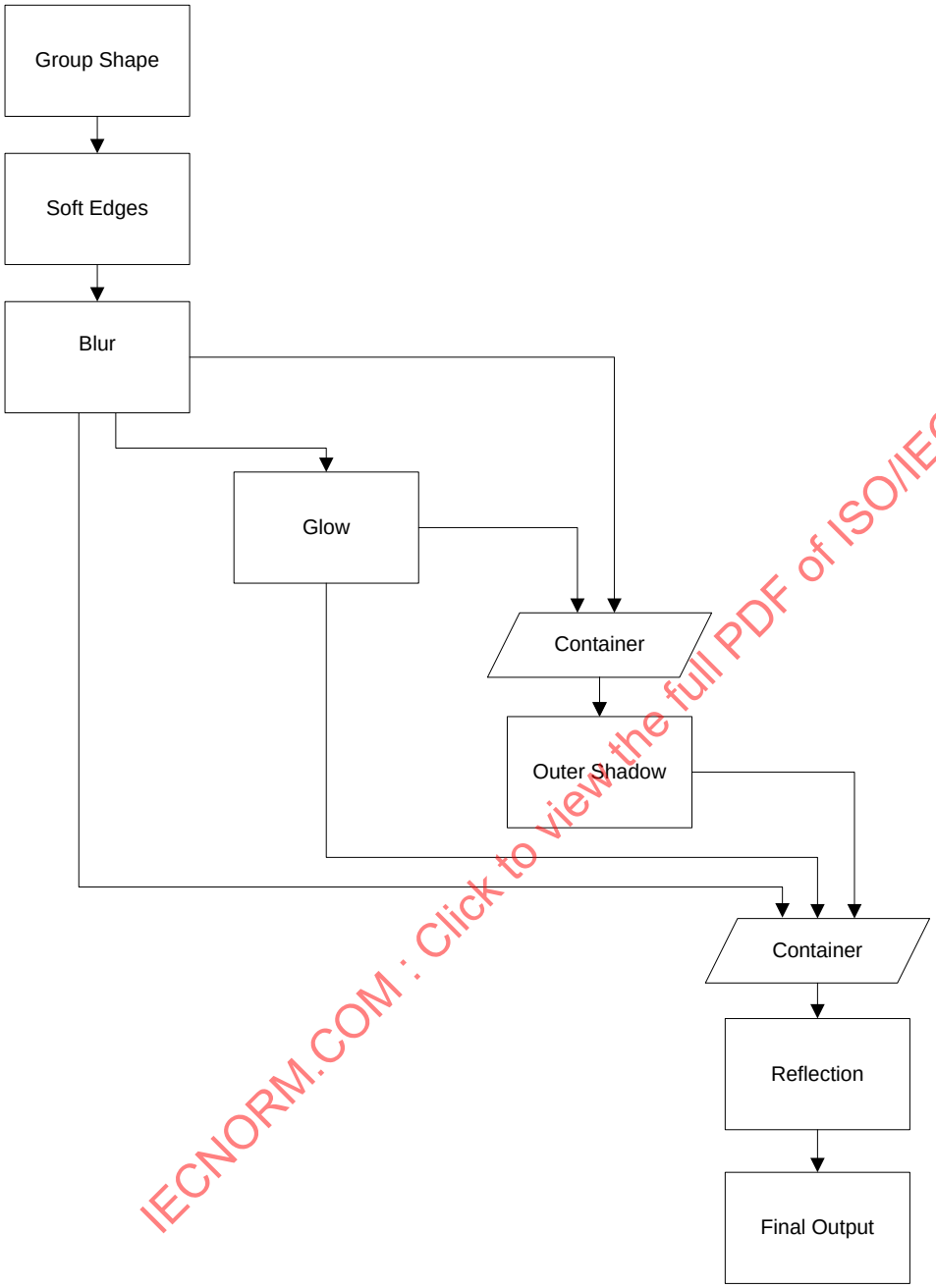
effectLst Processing for Shapes



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

effectLst Processing for Group Shapes



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectList) is located in §A.4.1. *end note*]

20.1.8.27 fgClr (Foreground color)

This element specifies the foreground color of a pattern fill.

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

20.1.8.28 fill (Fill)

This element specifies a fill which is one of blipFill, gradFill, grpFill, noFill, pattFill or solidFill.

[*Note*: The W3C XML Schema definition of this element's content model (CT_FillEffect) is located in §A.4.1. *end note*]

20.1.8.29 fillOverlay (Fill Overlay Effect)

This element specifies a fill overlay effect. A fill overlay can be used to specify an additional fill for an object and blend the two fills together.

Attributes	Description
blend (Blend)	Specifies how to blend the fill with the base effect. The possible values for this attribute are defined by the ST_BlendMode simple type (§20.1.10.11).

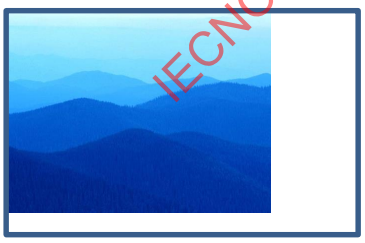
[*Note*: The W3C XML Schema definition of this element's content model (CT_FillOverlayEffect) is located in §A.4.1. *end note*]

20.1.8.30 fillRect (Fill Rectangle)

This element specifies a fill rectangle. When stretching of an image is specified, a source rectangle, srcRect, is scaled to fit the specified fill rectangle.

Each edge of the fill rectangle is defined by a percentage offset from the corresponding edge of the shape's bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note*: For example, a left offset of 25% specifies that the left edge of the fill rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

[*Example*:



```
<a:blipFill>
  <a:blip r:embed="rId2"/>
  <a:stretch>
      <a:fillRect b="10000" r="25000"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </a:stretch>
  </a:blipFill>
```

The above image is stretched to fill the entire rectangle except for the bottom 10% and rightmost 25%.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

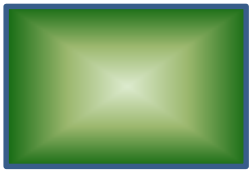
[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.31 fillToRect (Fill To Rectangle)

This element defines the "focus" rectangle for the center shade, specified relative to the fill tile rectangle. The center shade fills the entire tile except the margins specified by each attribute.

Each edge of the center shade rectangle is defined by a percentage offset from the corresponding edge of the tile rectangle. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note:* For example, a left offset of 25% specifies that the left edge of the center shade rectangle is located to the right of the tile rectangle's left edge by an amount equal to 25% of the tile rectangle's width. *end note*]

[*Example*:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:path path="rect">
  <a:fillToRect l="50000" r="50000" t="50000" b="50000"/>
</a:path>
```

In the above shape, the rectangle defined by fillToRect is a single point in the center of the shape. This creates the effect of the center shade focusing at a point in the center of the region.

end example]

[*Example*:

```
<a:path path="rect">
  <a:fillToRect l="25000" t="25000" r="25000" b="0"/>
</a:path>
```

The center shade occupies the rectangle defined by excluding the topmost, leftmost, and rightmost 25% of the region. Therefore, the gradient fills the remaining leftmost 25%, topmost 25%, and rightmost 25% of the region.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.8.32 glow (Glow Effect)

This element specifies a glow effect, in which a color blurred outline is added outside the edges of the object.

Attributes	Description
rad (Radius)	Specifies the radius of the glow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GlowEffect) is located in §A.4.1. *end note*]

20.1.8.33 gradFill (Gradient Fill)

This element defines a gradient fill.

A gradient fill is a fill which is characterized by a smooth gradual transition from one color to the next. At its simplest, it is a fill which transitions between two colors; or more generally, it can be a transition of any number of colors.

The desired transition colors and locations are defined in the gradient stop list (gsLst) child element.

The other child element defines the properties of the gradient fill (there are two styles-- a linear shade style as well as a path shade style)

[*Example*:

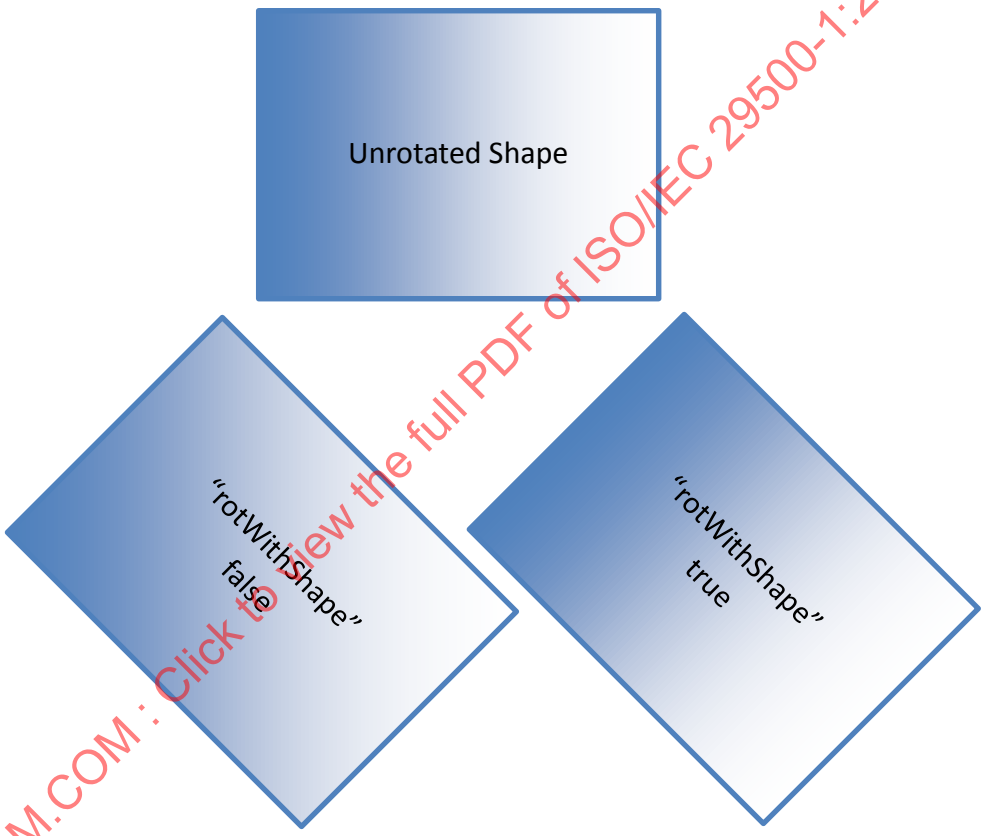
The following is a sample gradient fill, varying from blue to white:

end example]

Attributes	Description
flip (Tile Flip)	Specifies the direction(s) in which to flip the gradient while tiling. Normally a gradient fill encompasses the entire bounding box of the shape which contains the fill. However, with the tileRect element, it is possible to define a "tile" rectangle which is smaller than the bounding box. In this situation, the gradient fill is encompassed within the tile rectangle, and the tile rectangle is tiled across the bounding box to fill the entire area.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TileFlipMode simple type (§20.1.10.86).
rotWithShape (Rotate With Shape)	Specifies if a fill rotates along with a shape when the shape is rotated. [*Example*: The following is a fill with the flip attribute set to "x". The black interior rectangle indicates the tile rectangle. Notice that the adjacent rectangle to the right in the tile has been flipped along the x-axis.  *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientFillProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.8.34 grayscl (Gray Scale Effect)

This element specifies a gray scale effect. Converts all effect color values to a shade of gray, corresponding to their luminance. Effect alpha (opacity) values are unaffected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GrayscaleEffect) is located in §A.4.1. *end note*]

20.1.8.35 grpFill (Group Fill)

This element specifies a group fill. When specified, this setting indicates that the parent element is part of a group and should inherit the fill properties of the group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupFillProperties) is located in §A.4.1. *end note*]

20.1.8.36 gs (Gradient stops)

This element defines a gradient stop. A gradient stop consists of a position where the stop appears in the color band.

Attributes	Description
pos (Position)	Specifies where this gradient stop should appear in the color band. This position is specified in the range [0%, 100%], which corresponds to the beginning and the end of the color band respectively. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientStop) is located in §A.4.1. *end note*]

20.1.8.37 gsLst (Gradient Stop List)

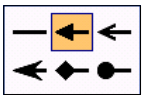
The list of gradient stops that specifies the gradient colors and their relative positions in the color band.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GradientStopList) is located in §A.4.1. *end note*]

20.1.8.38 headEnd (Line Head/End Style)

This element specifies decorations which can be added to the head of a line.

[*Example*:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end exmaple]

Attributes	Description
len (Length of Head/End)	Specifies the line end length in relation to the line width. The possible values for this attribute are defined by the ST_LineEndLength simple type (§20.1.10.32).
type (Line Head/End Type)	Specifies the line end decoration, such as a triangle or arrowhead. The possible values for this attribute are defined by the ST_LineEndType simple type (§20.1.10.33).
w (Width of Head/End)	Specifies the line end width in relation to the line width. The possible values for this attribute are defined by the ST_LineEndWidth simple type (§20.1.10.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineEndProperties) is located in §A.4.1. *end note*]

20.1.8.39 hsl (Hue Saturation Luminance Effect)

This element specifies a hue/saturation/luminance effect. The hue, saturation, and luminance can each be adjusted relative to its current value.

Attributes	Description
hue (Hue)	Specifies the number of degrees by which the hue is adjusted. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
lum (Luminance)	Specifies the percentage by which the luminance is adjusted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
sat (Saturation)	Specifies the percentage by which the saturation is adjusted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HSLEffect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.8.40 innerShdw (Inner Shadow Effect)

This element specifies an inner shadow effect. A shadow is applied within the edges of the object according to the parameters given by the attributes.

Attributes	Description
blurRad (Blur Radius)	Specifies the blur radius. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
dist (Distance)	Specifies how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_InnerShadowEffect) is located in §A.4.1. *end note*]

20.1.8.41 lin (Linear Gradient Fill)

This element specifies a linear gradient.

Attributes	Description
ang (Angle)	Specifies the direction of color change for the gradient. To define this angle, let its value be x measured clockwise. Then (-sin x, cos x) is a vector parallel to the line of constant color in the gradient fill. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
scaled (Scaled)	Whether the gradient angle scales with the fill region. Mathematically, if this flag is true, then the gradient vector (cos x , sin x) is scaled by the width (w) and height (h) of the fill region, so that the vector becomes (w cos x, h sin x) (before normalization). Observe that now if the gradient angle is 45 degrees, the gradient vector is (w, h), which goes

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	from top-left to bottom-right of the fill region. If this flag is false, the gradient angle is independent of the fill region and is not scaled using the manipulation described above. So a 45-degree gradient angle always give a gradient band whose line of constant color is parallel to the vector (1, -1). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LinearShadeProperties) is located in §A.4.1. *end note*]

20.1.8.42 lum (Luminance Effect)

This element specifies a luminance effect. Brightness linearly shifts all colors closer to white or black. Contrast scales all colors to be either closer or further apart.

Attributes	Description
bright (Brightness)	Specifies the percent to change the brightness. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
contrast (Contrast)	Specifies the percent to change the contrast. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LuminanceEffect) is located in §A.4.1. *end note*]

20.1.8.43 miter (Miter Line Join)

This element specifies that a line join shall be mitered.

[*Example*: The following sample illustrated two lines which are joined using a mitered style

end example]

Attributes	Description
lim (Miter Join	Specifies the amount by which lines is extended to form a miter join - otherwise miter

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Limit)	joins can extend infinitely far (for lines which are almost parallel). The possible values for this attribute are defined by the ST_PositivePercentage simple type (§20.1.10.46).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinMiterProperties) is located in §A.4.1. *end note*]

20.1.8.44 noFill (No Fill)

This element specifies that no fill is applied to the parent element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NoFillProperties) is located in §A.4.1. *end note*]

20.1.8.45 outerShdw (Outer Shadow Effect)

This element specifies an Outer Shadow Effect.

[*Example*: The following is an example of an outer shadow effect.

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
algn (Shadow Alignment)	Specifies shadow alignment; alignment happens first, effectively setting the origin for scale, skew, and offset. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.53).
blurRad (Blur Radius)	Specifies the blur radius of the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Shadow Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	type (§20.1.10.44).
dist (Shadow Offset Distance)	Specifies the how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
kx (Horizontal Skew)	Specifies the horizontal skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
rotWithShape (Rotate With Shape)	Specifies whether the shadow rotates with the shape if the shape is rotated. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
sx (Horizontal Scaling Factor)	Specifies the horizontal scaling factor; negative scaling causes a flip. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Scaling Factor)	Specifies the vertical scaling factor; negative scaling causes a flip. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

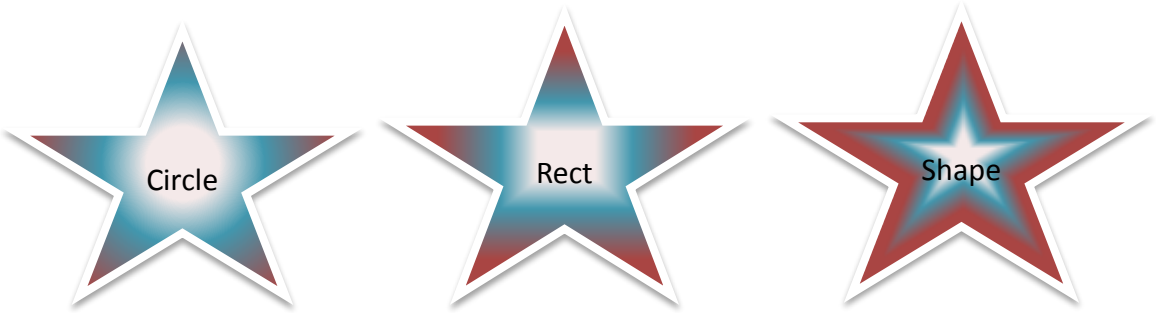
[*Note*: The W3C XML Schema definition of this element's content model (CT_OuterShadowEffect) is located in §A.4.1. *end note*]

20.1.8.46 path (Path Gradient)

This element defines that a gradient fill follows a path vs. a linear line.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



The examples above illustrate gradient fills following a circular, rectangular or shape path.

end example]

Attributes	Description
path (Gradient Fill Path)	Specifies the shape of the path to follow. The possible values for this attribute are defined by the ST_PathShadeType simple type (§20.1.10.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PathShadeProperties) is located in §A.4.1. *end note*]

20.1.8.47 pattFill (Pattern Fill)

This element specifies a pattern fill. A repeated pattern is used to fill the object.

Attributes	Description
prst (Preset Pattern)	Specifies one of a set of preset patterns to fill the object. The possible values for this attribute are defined by the ST_PresetPatternVal simple type (§20.1.10.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PatternFillProperties) is located in §A.4.1. *end note*]

20.1.8.48 prstDash (Preset Dash)

This element specifies that a preset line dashing scheme should be used.

Attributes	Description
val (Value)	Specifies which preset dashing scheme is to be used. The possible values for this attribute are defined by the ST_PresetLineDashVal simple type (§20.1.10.49).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetLineDashProperties) is located in §A.4.1. *end note*]

20.1.8.49 prstShdw (Preset Shadow)

This element specifies that a preset shadow is to be used. Each preset shadow is equivalent to a specific outer shadow effect. For each preset shadow, the color element, direction attribute, and distance attribute represent the color, direction, and distance parameters of the corresponding outer shadow. Additionally, the rotateWithShape attribute of corresponding outer shadow is always false. Other non-default parameters of the outer shadow are dependent on the prst attribute.

Attributes	Description
dir (Direction)	Specifies the direction to offset the shadow. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
dist (Distance)	Specifies how far to offset the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prst (Preset Shadow)	Specifies which preset shadow to use. The possible values for this attribute are defined by the ST_PresetShadowVal simple type (§20.1.10.52).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetShadowEffect) is located in §A.4.1. *end note*]

20.1.8.50 reflection (Reflection Effect)

This element specifies a reflection effect.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
algn (Shadow Alignment)	Specifies shadow alignment. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.53).
blurRad (Blur Radius)	Specifies the blur radius. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
dir (Direction)	Specifies the direction of the alpha gradient ramp relative to the shape itself. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
dist (Distance)	Specifies how far to distance the shadow. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
endA (End Alpha)	Specifies the ending reflection opacity.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).
endPos (End Position)	Specifies the end position (along the alpha gradient ramp) of the end alpha value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).
fadeDir (Fade Direction)	Specifies the direction to offset the reflection. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).
kx (Horizontal Skew)	Specifies the horizontal skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
rotWithShape (Rotate With Shape)	Specifies if the reflection rotates with the shape. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
stA (Start Opacity)	starting reflection opacity. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).
stPos (Start Position)	Specifies the start position (along the alpha gradient ramp) of the start alpha value. The possible values for this attribute are defined by the ST_PositiveFixedPercentage simple type (§20.1.10.45).
sx (Horizontal Ratio)	Specifies the horizontal scaling factor. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies the vertical scaling factor. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ReflectionEffect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.8.51 relOff (Relative Offset Effect)

This element specifies a relative offset effect. Sets up a new origin by offsetting relative to the size of the previous effect.

Attributes	Description
tx (Offset X)	Specifies the X offset. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
ty (Offset Y)	Specifies the Y offset. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeOffsetEffect) is located in §A.4.1. *end note*]

20.1.8.52 round (Round Line Join)

This element specifies that lines joined together have a round join.

[*Example*:

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineJoinRound) is located in §A.4.1. *end note*]

20.1.8.53 softEdge (Soft Edge Effect)

This element specifies a soft edge effect. The edges of the shape are blurred, while the fill is not affected.

Attributes	Description
rad (Radius)	Specifies the radius of blur to apply to the edges. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SoftEdgesEffect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.8.54 solidFill (Solid Fill)

This element specifies a solid color fill. The shape is filled entirely with the specified color.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SolidColorFillProperties) is located in §A.4.1. *end note*]

20.1.8.55 srcRect (Source Rectangle)

This element specifies the portion of the blip used for the fill.

Each edge of the source rectangle is defined by a percentage offset from the corresponding edge of the bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note:* For example, a left offset of 25% specifies that the left edge of the source rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.56 stretch (Stretch)

This element specifies that a BLIP should be stretched to fill the target rectangle. The other option is a tile where a BLIP is tiled to fill the available area.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StretchInfoProperties) is located in §A.4.1. *end note*]

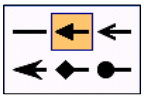
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.8.57 tailEnd (Tail line end style)

This element specifies decorations which can be added to the tail of a line.

[*Example*:



end example]

Attributes	Description
len (Length of Head/End)	Specifies the line end length in relation to the line width. The possible values for this attribute are defined by the ST_LineEndLength simple type (§20.1.10.32).
type (Line Head/End Type)	Specifies the line end decoration, such as a triangle or arrowhead. The possible values for this attribute are defined by the ST_LineEndType simple type (§20.1.10.33).
w (Width of Head/End)	Specifies the line end width in relation to the line width. The possible values for this attribute are defined by the ST_LineEndWidth simple type (§20.1.10.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineEndProperties) is located in §A.4.1. *end note*]

20.1.8.58 tile (Tile)

This element specifies that a BLIP should be tiled to fill the available space. This element defines a "tile" rectangle within the bounding box. The image is encompassed within the tile rectangle, and the tile rectangle is tiled across the bounding box to fill the entire area.

[*Example*:

The following is a fill with the flip attribute set to "x". The black interior rectangle indicates the tile rectangle. Notice that the adjacent rectangle to the right in the tile has been flipped along the x-axis.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
algn (Alignment)	Specifies where to align the first tile with respect to the shape. Alignment happens after the scaling, but before the additional offset. The possible values for this attribute are defined by the ST_RectAlignment simple type (§20.1.10.53).
flip (Tile Flipping)	Specifies the direction(s) in which to flip the source image while tiling. Images can be flipped horizontally, vertically, or in both directions to fill the entire region. The possible values for this attribute are defined by the ST_TileFlipMode simple type (§20.1.10.86).
sx (Horizontal Ratio)	Specifies the amount to horizontally scale the srcRect. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies the amount to vertically scale the srcRect. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
tx (Horizontal Offset)	Specifies additional horizontal offset after alignment. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
ty (Vertical Offset)	Specifies additional vertical offset after alignment. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TileInfoProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.8.59 tileRect (Tile Rectangle)

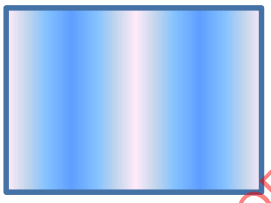
This element specifies a rectangular region of the shape to which the gradient is applied. This region is then tiled across the remaining area of the shape to complete the fill. The tile rectangle is defined by percentage offsets from the sides of the shape's bounding box.

Each edge of the tile rectangle is defined by a percentage offset from the corresponding edge of the bounding box. A positive percentage specifies an inset, while a negative percentage specifies an outset. [*Note*: For example, a left offset of 25% specifies that the left edge of the tile rectangle is located to the right of the bounding box's left edge by an amount equal to 25% of the bounding box's width. *end note*]

[*Example*:

The image above depicts a horizontal gradient with no tileRect element.

The image above depicts the same gradient with a tileRect element specifying l="50000" (50%). The right half of the shape is the tile to which the gradient is applied, and the left half of the shape contains a tiled copy of that gradient fill.



The image above depicts the same gradient with a tileRect element specifying l="75000" (75%). The rightmost 25% of the shape contains the tile rectangle to which the gradient is applied. This gradient is tiled three times to cover the leftmost 75% of the shape. The tile rectangle is flipped horizontally when covering the shape.

end example]

Attributes	Description
b (Bottom Offset)	Specifies the bottom edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§20.1.10.40).
l (Left Offset)	Specifies the left edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
r (Right Offset)	Specifies the right edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
t (Top Offset)	Specifies the top edge of the rectangle. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelativeRect) is located in §A.4.1. *end note*]

20.1.8.60 tint (Tint Effect)

This element specifies a tint effect. Shifts effect color values towards/away from hue by the specified amount.

Attributes	Description
amt (Amount)	Specifies by how much the color value is shifted. The possible values for this attribute are defined by the ST_FixedPercentage simple type (§20.1.10.24).
hue (Hue)	Specifies the hue towards which to tint. The possible values for this attribute are defined by the ST_PositiveFixedAngle simple type (§20.1.10.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TintEffect) is located in §A.4.1. *end note*]

20.1.8.61 xfrm (Transform Effect)

This element specifies a transform effect. The transform is applied to each point in the shape's geometry using the following matrix:

$$\begin{bmatrix} sx & \tan(kx) & tx \\ \tan(ky) & sy & ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
kx (Horizontal Skew)	Specifies the horizontal skew angle, defined as the angle between the top-left corner and bottom-left corner of the object's original bounding box. If positive, the bottom edge of the shape is positioned to the right relative to the top edge. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
ky (Vertical Skew)	Specifies the vertical skew angle, defined as the angle between the top-left corner and top-right corner of the object's original bounding box. If positive, the right edge of the object is positioned lower relative to the left edge. The possible values for this attribute are defined by the ST_FixedAngle simple type (§20.1.10.23).
sx (Horizontal Ratio)	Specifies a percentage by which to horizontally scale the object. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
sy (Vertical Ratio)	Specifies a percentage by which to vertically scale the object. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
tx (Horizontal Shift)	Specifies an amount by which to shift the object along the x-axis. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
ty (Vertical Shift)	Specifies an amount by which to shift the object along the y-axis. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TransformEffect) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.9 Shape Definitions and Attributes

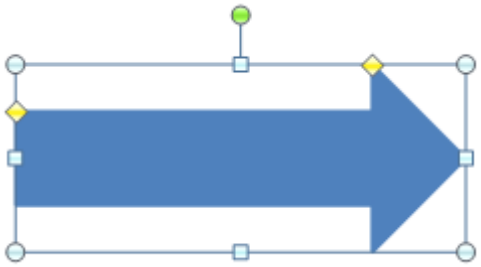
The Shape Definitions and Attributes portion of the DrawingML framework deals with all geometric properties for shapes within a document. This includes both preset geometries that publicly are interpreted by the generating application and custom geometries that have their points and curves explicitly specified. In addition to the underlying geometry of the shape there are also other coordinate-based properties for each shape that this framework describes.

©ISO/IEC 2016 – All rights reserved

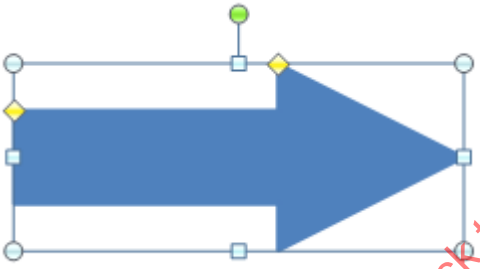
20.1.9.1 ahLst (List of Shape Adjust Handles)

This element specifies the adjust handles that are applied to a custom geometry. These adjust handles specify points within the geometric shape that can be used to perform certain transform operations on the shape.

[*Example*: Consider the scenario where a custom geometry, an arrow in this case, has been drawn and adjust handles have been placed at the top left corner of both the arrow head and arrow body. The user interface can then be made to transform only certain parts of the shape by using the corresponding adjust handle.



For instance if the user wished to change only the width of the arrow head then they would use the adjust handle located on the top left of the arrow head. The result of adjusting this transforms the shape as shown below.



end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjustHandleList) is located in §A.4.1. *end note*]

20.1.9.2 ahPolar (Polar Adjust Handle)

This element specifies a polar adjust handle for a custom shape. The position of this adjust handle is specified by the corresponding pos child element. The allowed adjustment of this adjust handle are specified via it's min and max attributes. Based on the adjustment of this adjust handle certain corresponding guides are updated to contain these values.

Attributes	Description
gdRefAng (Angle Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment angle from this adjust handle.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
gdRefR (Radial Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment radius from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
maxAng (Maximum Angle Adjustment)	Specifies the maximum angle position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move angularly. That is the maxAng and minAng are equal. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
maxR (Maximum Radial Adjustment)	Specifies the maximum radial position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move radially. That is the maxR and minR are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minAng (Minimum Angle Adjustment)	Specifies the minimum angle position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move angularly. That is the maxAng and minAng are equal. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
minR (Minimum Radial Adjustment)	Specifies the minimum radial position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move radially. That is the maxR and minR are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PolarAdjustHandle) is located in §A.4.1. *end note*]

20.1.9.3 ahXY (XY Adjust Handle)

This element specifies an XY-based adjust handle for a custom shape. The position of this adjust handle is specified by the corresponding pos child element. The allowed adjustment of this adjust handle are specified via it's min and max type attributes. Based on the adjustment of this adjust handle certain corresponding guides are updated to contain these values.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
gdRefX (Horizontal Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment x position from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
gdRefY (Vertical Adjustment Guide)	Specifies the name of the guide that is updated with the adjustment y position from this adjust handle. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).
maxX (Maximum Horizontal Adjustment)	Specifies the maximum horizontal position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the x direction. That is the maxX and minX are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
maxY (Maximum Vertical Adjustment)	Specifies the maximum vertical position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the y direction. That is the maxY and minY are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minX (Minimum Horizontal Adjustment)	Specifies the minimum horizontal position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the x direction. That is the maxX and minX are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
minY (Minimum Vertical Adjustment)	Specifies the minimum vertical position that is allowed for this adjustment handle. If this attribute is omitted, then it is assumed that this adjust handle cannot move in the y direction. That is the maxY and minY are equal. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_XYAdjustHandle) is located in §A.4.1. *end note*]

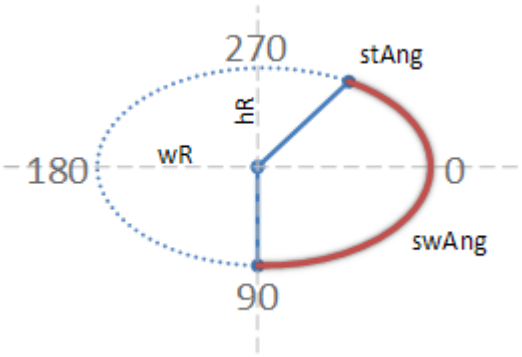
20.1.9.4 arcTo (Draw Arc To)

This element specifies the existence of an arc within a shape path. It draws an arc with the specified parameters from the current pen position to the new point specified. An arc is a line that is bent based on the shape of a

©ISO/IEC 2016 – All rights reserved

supposed circle. The length of this arc is determined by specifying both a start angle and an ending angle that act together to effectively specify an end point for the arc.

[*Example*: The diagram shown below represents a single arc that has a start angle of 300 degrees and a swing angle of 150 degrees. This arc is drawn using the supposed circle that is described using the hR and wR attributes as shown below. The degrees by which the stAng must abide is shown along the circumference of the circle. These degrees are to be specified in 60,000ths of a degree. If this arc were part of a shape the start angle point along the circle would be the starting point along the path and the ending point would be the ending of the angle swing along this supposed circle. That is any shape geometry coming before this arc in the shape path would be joined with the upper point of this arc and consequently any geometry coming after this arc in the path would be joined with the lower point of this arc.



end example]

Attributes	Description
hR (Shape Arc Height Radius)	This attribute specifies the height radius of the supposed circle being used to draw the arc. This gives the circle a total height of (2 * hR). This total height could also be called it's vertical diameter as it is the diameter for the y axis only. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
stAng (Shape Arc Start Angle)	Specifies the start angle for an arc. This angle specifies what angle along the supposed circle path is used as the start position for drawing the arc. This start angle is locked to the last known pen position in the shape path. Thus guaranteeing a continuos shape path. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).
swAng (Shape Arc Swing Angle)	Specifies the swing angle for an arc. This angle specifies how far angle-wise along the supposed cicle path the arc is extended. The extension from the start angle is always in the clockwise direction around the supposed circle. The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

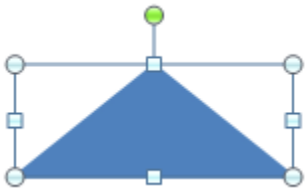
Attributes	Description
wR (Shape Arc Width Radius)	This attribute specifies the width radius of the supposed circle being used to draw the arc. This gives the circle a total width of (2 * wR). This total width could also be called it's horizontal diameter as it is the diameter for the x axis only. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DArcTo) is located in §A.4.1. *end note*]

20.1.9.5 avLst (List of Shape Adjust Values)

This element specifies the adjust values that are applied to the specified shape. An adjust value is simply a guide that has a value based formula specified. That is, no calculation takes place for an adjust value guide. Instead, this guide specifies a parameter value that is used for calculations within the shape guides.

[*Example*: Consider the case where the user would like to specify a triangle with it's bottom edge defined not by static points but by using a varying parameter, namely an adjust value. Consider the diagrams and DrawingML shown below. This first triangle has been drawn with a bottom edge that is equal to the height, namely 2. Thus we see in the figure below that the bottom of the triangle matches the bottom of the shape bounding box.



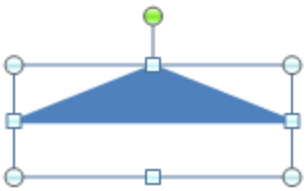
```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst>
    <a:gd name="myGuide" fmla="val 2"/>
  </a:avLst>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <a:pathLst>
      <a:path w="2" h="2">
        <a:moveTo>
          <a:pt x="0" y="myGuide"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="2" y="myGuide"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="1" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

If however we change the adjust value to half that, namely 1. Then we see the entire bottom edge of the triangle move to now be placed along the vertical midpoint within the shape bounding box. This is because both of the bottom points in this triangle depend on this adjust value for their coordinate positions. The triangle and corresponding DrawingML shown below illustrate this point.



```
  <a:avLst>
    <a:gd name="myGuide" fmla="val 1"/>
  </a:avLst>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuideList) is located in §A.4.1. *end note*]

20.1.9.6 close (Close Shape Path)

This element specifies the ending of a series of lines and curves in the creation path of a custom geometric shape. When this element is encountered, the generating application should consider the corresponding path closed. That is, any further lines or curves that follow this element should be ignored.

[*Note*: A path can be specified and not closed. A path such as this cannot however have any fill associated with it as it has not been considered a closed geometric path. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

[*Note*: When the last point in the creation path does not meet with the first point in the creation path the generating application should connect the last point with the first via a straight line, thus creating a closed shape geometry. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DClose) is located in §A.4.1. *end note*]

20.1.9.7 cubicBezTo (Draw Cubic Bezier Curve To)

This element specifies to draw a cubic bezier curve along the specified points. To specify a cubic bezier curve there needs to be 3 points specified. The first two are control points used in the cubic bezier calculation and the last is the ending point for the curve. The coordinate system used for this kind of curve is the path coordinate system as this element is path specific.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DCubicBezierTo) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

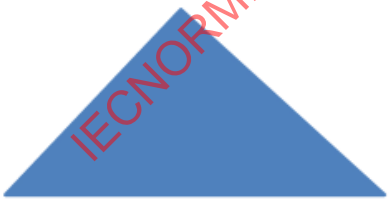
©ISO/IEC 2016 – All rights reserved

20.1.9.8 custGeom (Custom Geometry)

This element specifies the existence of a custom geometric shape. This shape consists of a series of lines and curves described within a creation path. In addition to this there can also be adjust values, guides, adjust handles, connection sites and an inscribed rectangle specified for this custom geometric shape.

[*Example*: Consider the scenario when a preset geometry does not accurately depict what must be displayed in the document. For this a custom geometry can be used to define most any 2-dimensional geometric shape. Shown below is an example of such a custom geometry.

```
<a:custGeom>
  <a:avLst/>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2650602" h="1261641">
      <a:moveTo>
        <a:pt x="0" y="1261641"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2650602" y="1261641"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1226916" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The custom geometry above is drawn by first moving to a specific starting point with the moveTo element. Then a series of lnTo elements in the creation path specify the lines that make up the borders of the shape and finally a close element is used to specify the end of the creation path. The resulting shape is shown above. *end example*]

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustomGeometry2D) is located in §A.4.1. *end note*]

20.1.9.9 cxn (Shape Connection Site)

This element specifies the existence of a connection site on a custom shape. A connection site allows a cxnSp to be attached to this shape. This connection is maintained when the shape is repositioned within the document. It should be noted that this connection is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

[*Note*: The transform coordinate system is different from a path coordinate system as it is per shape instead of per path within the shape. *end note*]

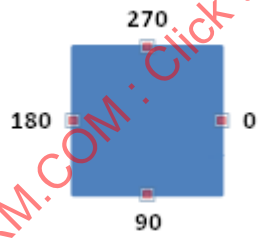
[*Example*: Consider the following custom geometry that has two connection sites specified. One connection is located at the bottom left of the shape and the other at the bottom right. The following DrawingML would describe such a custom geometry.

```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
  <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst>
    <a:cxn ang="0">
      <a:pos x="0" y="679622"/>
    </a:cxn>
    <a:cxn ang="0">
      <a:pos x="1705233" y="679622"/>
    </a:cxn>
  </a:cxnLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <a:rect l="0" t="0" r="0" b="0"/>
    <a:pathLst>
      <a:path w="2" h="2">
        <a:moveTo>
          <a:pt x="0" y="2"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="2" y="2"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="1" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

end example]

Attributes	Description
ang (Connection Site Angle)	Specifies the incoming connector angle. This angle is the angle around the connection site that an incoming connector tries to be routed to. This allows connectors to know where the shape is in relation to the connection site and route connectors so as to avoid any overlap with the shape. [*Example*: Consider a simple square. In order to not have any connectors routed over the shape, the collowing angles would be specified for their respective connection sites.  *end example*] The possible values for this attribute are defined by the ST_AdjAngle simple type (§20.1.10.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectionSite) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.9.10 cxnLst (List of Shape Connection Sites)

This element specifies all the connection sites that are used for this shape. A connection site is specified by defining a point within the shape bounding box that can have a cxnSp element attached to it. These connection sites are specified using the shape coordinate system that is specified within the ext transform element.

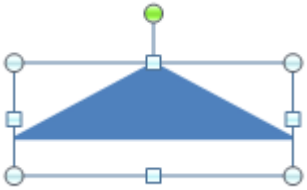
[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectionSiteList) is located in §A.4.1. *end note*]

20.1.9.11 gd (Shape Guide)

This element specifies the precense of a shape guide that is used to govern the geometry of the specified shape. A shape guide consists of a formula and a name that the result of the formula is assigned to. Recognized formulas are listed with the fmla attribute documentation for this element.

[*Note*: The order in which guides are specified determines the order in which their values are calculated. For instance it is not possible to specify a guide that uses another guides result when that guide has not yet been calculated. *end note*]

[*Example*: Consider the case where the user would like to specify a triangle with it's bottom edge defined not by static points but by using a varying parameter, namely an guide. Consider the diagrams and DrawingML shown below. This first triangle has been drawn with a bottom edge that is equal to the 2/3 the value of the shape height. Thus we see in the figure below that the triangle appears to occupy 2/3 of the vertical space within the shape bounding box.

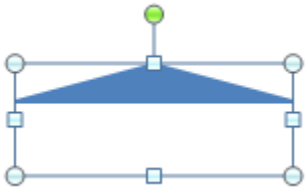


```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
  <a:gdLst>
    <a:gd name="myGuide" fmla="*/ h 2 3"/>
  </a:gdLst>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <a:path w="1705233" h="679622">
        <a:moveTo>
          <a:pt x="0" y="myGuide"/>
        </a:moveTo>
        <a:lnTo>
          <a:pt x="1705233" y="myGuide"/>
        </a:lnTo>
        <a:lnTo>
          <a:pt x="852616" y="0"/>
        </a:lnTo>
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

If however we change the guide to half that, namely 1/3. Then we see the entire bottom edge of the triangle move to now only occupy 1/3 of the toal space within the shape bounding box. This is because both of the bottom points in this triangle depend on this guide for their coordinate positions. The triangle and corresponding DrawingML shown below illustrate this point.



```
  <a:gdLst>
    <a:gd name="myGuide" fmla="*/ h 1 3"/>
  </a:gdLst>
```

end example]

Attributes	Description
fmla (Shape Guide Formula)	Specifies the formula that is used to calculate the value for a guide. Each formula has a certain number of arguments and a specific set of operations to perform on these arguments in order to generate a value for a guide. There are a total of 17 different formulas available. These are shown below with the usage for each defined. **('*/') - Multiply Divide Formula** Arguments: 3 (fmla="*/ x y z") Usage: "*/ x y z" = ((x * y) / z) = value of this guide **('+-') - Add Subtract Formula** Arguments: 3 (fmla="+- x y z") Usage: "+- x y z" = ((x + y) - z) = value of this guide

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	('+/') - Add Divide Formula Arguments: 3 (fmla="+/ x y z") Usage: "+/ x y z" = ((x + y) / z) = value of this guide **('?:') - If Else Formula** Arguments: 3 (fmla="?: x y z") Usage: "?: x y z" = if (x > 0), then y = value of this guide, else z = value of this guide **('abs') - Absolute Value Formula** Arguments: 1 (fmla="abs x") Usage: "abs x" = if (x < 0), then (-1) * x = value of this guide else x = value of this guide **('at2') - ArcTan Formula** Arguments: 2 (fmla="at2 x y") Usage: "at2 x y" = arctan(y / x) = value of this guide **('cat2') - Cosine ArcTan Formula** Arguments: 3 (fmla="cat2 x y z") Usage: "cat2 x y z" = (x*(cos(arctan(z / y))) = value of this guide **('cos') - Cosine Formula** Arguments: 2 (fmla="cos x y") Usage: "cos x y" = (x * cos(y)) = value of this guide **('max') - Maximum Value Formula** Arguments: 2 (fmla="max x y") Usage: "max x y" = if (x > y), then x = value of this guide else y = value of this guide **('min') - Minimum Value Formula** Arguments: 2 (fmla="min x y") Usage: "min x y" = if (x < y), then x = value of this guide else y = value of this guide **('mod') - Modulo Formula** Arguments: 3 (fmla="mod x y z") Usage: "mod x y z" = sqrt(x^2 + b^2 + c^2) = value of this guide **('pin') - Pin To Formula** Arguments: 3 (fmla="pin x y z") Usage: "pin x y z" = if (y < x), then x = value of this guide else if (y > z), then z = value of this guide else y = value of this guide **('sat2') - Sine ArcTan Formula** Arguments: 3 (fmla="sat2 x y z") Usage: "sat2 x y z" = (x*sin(arctan(z / y))) = value of this guide **('sin') - Sine Formula** Arguments: 2 (fmla="sin x y")

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Usage: "sin x y" = (x * sin(y)) = value of this guide **('sqrt') - Square Root Formula** Arguments: 1 (fmla="sqrt x") Usage: "sqrt x" = sqrt(x) = value of this guide **('tan') - Tangent Formula** Arguments: 2 (fmla="tan x y") Usage: "tan x y" = (x * tan(y)) = value of this guide **('val') - Literal Value Formula** Arguments: 1 (fmla="val x") Usage: "val x" = x = value of this guide [*Note*: Guides that have a literal value formula specified via fmla="val x" above should only be used within the avLst as an adjust value for the shape. This however is not strictly enforced. *end note*] The possible values for this attribute are defined by the ST_GeomGuideFormula simple type (§20.1.10.27).
name (Shape Guide Name)	Specifies the name that is used to reference to this guide. This name can be used just as a variable would within an equation. That is this name can be substituted for literal values within other guides or the specification of the shape path. The possible values for this attribute are defined by the ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuide) is located in §A.4.1. *end note*]

20.1.9.12 gdLst (List of Shape Guides)

This element specifies all the guides that are used for this shape. A guide is specified by the gd element and defines a calculated value that can be used for the construction of the corresponding shape.

[*Note*: Guides that have a literal value formula specified via fmla="val x" above should only be used within the avLst as an adjust value for the shape. This however is not strictly enforced. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomGuideList) is located in §A.4.1. *end note*]

20.1.9.13 lnTo (Draw Line To)

This element specifies the drawing of a straight line from the current pen position to the new point specified. This line becomes part of the shape geometry, representing a side of the shape. The coordinate system used when specifying this line is the path coordinate system.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DLineTo) is located in §A.4.1. *end note*]

20.1.9.14 moveTo (Move Path To)

This element specifies a set of new coordinates to move the shape cursor to. This element is only used for drawing a custom geometry. When this element is utilized the pt element is used to specify a new set of shape coordinates that the shape cursor should be moved to. This does not draw a line or curve to this new position from the old position but simply move the cursor to a new starting position. It is only when a path drawing element such as lnTo is used that a portion of the path is drawn.

[*Example*: Consider the case where a user wishes to begin drawing a custom geometry not at the default starting coordinates of x=0 , y=0 but at coordinates further inset into the shape coordinate space. The following DrawingML would specify such a case.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

Notice the moveTo element advances the y coordinates before any actual lines are drawn. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DMoveTo) is located in §A.4.1. *end note*]

20.1.9.15 path (Shape Path)

This element specifies a creation path consisting of a series of moves, lines and curves that when combined forms a geometric shape. This element is only utilized if a custom geometry is specified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: Since multiple paths are allowed the rules for drawing are that the path specified later in the pathLst is drawn on top of all previous paths. *end note*]

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

Attributes	Description
extrusionOk (3D Extrusion Allowed)	Specifies that the use of 3D extrusions are possible on this path. This allows the generating application to know whether 3D extrusion can be applied in any form. If this attribute is omitted then a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fill (Path Fill)	Specifies how the corresponding path should be filled. If this attribute is omitted, a value of "norm" is assumed. The possible values for this attribute are defined by the ST_PathFillMode simple type (§20.1.10.37).
h (Path Height)	Specifies the height, or maximum y coordinate that should be used for within the path coordinate system. This value determines the vertical placement of all points within the corresponding path as they are all calculated using this height attribute as the max y

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	coordinate. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
stroke (Path Stroke)	Specifies if the corresponding path should have a path stroke shown. This is a boolean value that affect the outline of the path. If this attribute is omitted, a value of true is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
w (Path Width)	Specifies the width, or maximum x coordinate that should be used for within the path coordinate system. This value determines the horizontal placement of all points within the corresponding path as they are all calculated using this width attribute as the max x coordinate. The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2D) is located in §A.4.1. *end note*]

20.1.9.16 pathLst (List of Shape Paths)

This element specifies the entire path that is to make up a single geometric shape. The pathLst can consist of many individual paths within it.

[*Example*: Consider the following DrawingML.

```
<a:custGeom>
  <a:pathLst>
    <a:path w="2824222" h="590309">
      <a:moveTo>
        <a:pt x="0" y="428263"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="1620455" y="590309"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="2824222" y="173620"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1562582" y="0"/>
      </a:lnTo>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <a:close/>
      </a:path>
    </a:pathLst>
  </a:custGeom>
```

In the above example there is specified a four sided geometric shape that has all straight sides. While we only see three lines being drawn via the lnTo element there are actually four sides because the last point of (x=1562585, y=0) is connected to the first point in the creation path via a lnTo element. *end example*]

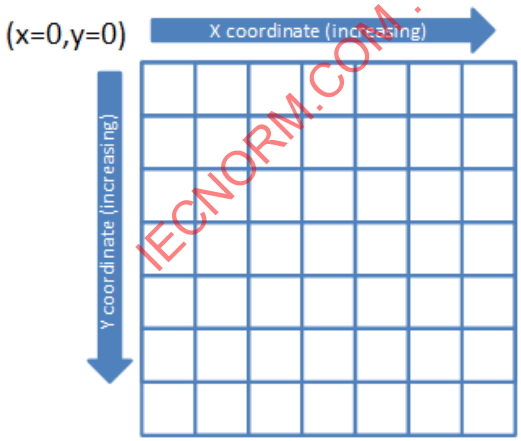
[*Note*: A geometry with multiple paths within it should be treated visually as if each path were a distinct shape. That is each creation path has its first point and last point joined to form a closed shape. However, the generating application should then connect the last point to the first point of the new shape. If a close element is encountered at the end of the previous creation path then this joining line should not be rendered by the generating application. The rendering should resume with the first line or curve on the new creation path. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DList) is located in §A.4.1. *end note*]

20.1.9.17 pos (Shape Position Coordinate)

Specifies a position coordinate within the shape bounding box. It should be noted that this coordinate is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

[*Note*: When specifying a point coordinate in path coordinate space it should be noted that the top left of the coordinate space is x=0, y=0 and the coordinate points for x grow to the right and for y grow down. This is illustrated in the diagram below.

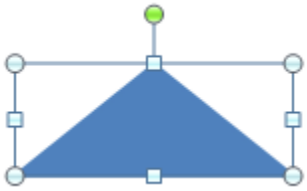


end note]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: To highlight the differences in the coordinate systems consider the drawing of the following triangle. Notice that the dimensions of the triangle are specified using the shape coordinate system with EMUs as the units via the ext transform element. Thus we see this shape is 1705233 EMUs wide by 679622 EMUs tall. However when looking at how the path for this shape is drawn we see that the x and y values fall between 0 and 2. This is because the path coordinate system has the arbitrary dimensions of 2 for the width and 2 for the height. Thus we see that a y coordinate of 2 within the path coordinate system specifies a y coordinate of 679622 within the shape coordinate system for this particular case.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
    <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2" h="2">
      <a:moveTo>
        <a:pt x="0" y="2"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2" y="2"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
x (X-Coordinate)	Specifies the x coordinate for this position coordinate. The units for this coordinate space are defined by the width of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
y (Y-Coordinate)	Specifies the y coordinate for this position coordinate. The units for this coordinate space are defined by the height of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjPoint2D) is located in §A.4.1. *end note*]

20.1.9.18 prstGeom (Preset geometry)

This element specifies when a preset geometric shape should be used instead of a custom geometric shape. The generating application should be able to render all preset geometries enumerated in the ST_ShapeType list.

[*Example*: Consider the scenario when a user does not wish to specify all the lines and curves that make up the desired shape but instead chooses to use a preset geometry. The following DrawingML would specify such a case.

```
<p:sp>
  <p:nvSpPr>
    <p:cNvPr id="4" name="My Preset Shape"/>
    <p:cNvSpPr/>
    <p:nvPr/>
  </p:nvSpPr>
  <p:spPr>
    <a:xfrm>
      <a:off x="1981200" y="533400"/>
      <a:ext cx="1143000" cy="1066800"/>
    </a:xfrm>
    <a:prstGeom prst="heart">
    </a:prstGeom>
  </p:spPr>
</p:sp>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The output shape rendered by this DrawingML is shown above. *end example*]

Attributes	Description

prst (Preset Shape)	Specifies the preset geometry that is used for this shape. This preset can have any of the values in the enumerated list for ST_ShapeType. This attribute is required in order for a preset geometry to be rendered.
	
	[*Example*: Consider the sample DrawingML below.
	
	`<p:sp>`
	`  <p:nvSpPr>`
	`  <p:cNvPr id="4" name="Sun 3"/>`
	`  <p:cNvSpPr/>`
	`  <p:nvPr/>`
	`  </p:nvSpPr>`
	`  <p:spPr>`
	`  <a:xfrm>`
	`  <a:off x="1981200" y="533400"/>`
	`  <a:ext cx="1143000" cy="1066800"/>`
	`  </a:xfrm>`
	`  <a:prstGeom prst="sun">`
	`  </a:prstGeom>`
	`  </p:spPr>`
	`</p:sp>`
	
	In the above example a preset geometry has been used to define a shape. The shape utilized here is the sun shape. *end example*]
	
	The possible values for this attribute are defined by the ST_ShapeType simple type (§20.1.10.56).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetGeometry2D) is located in §A.4.1. *end note*]

20.1.9.19 prstTxWarp (Preset Text Warp)

This element specifies when a preset geometric shape should be used to transform a piece of text. This operation is known formally as a text warp. The generating application should be able to render all preset geometries enumerated in the ST_TextShapeType list.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the case where the user wishes to accent a piece of text by warping it's shape. For this to occur a preset shape is chosen from the ST_TextShapeType list and applied to the entire body of text.

```
<p:sp>
  <p:txBody>
    <a:bodyPr wrap="none" rtlCol="0">
      <a:prstTxWarp prst="textInflate">
      </a:prstTxWarp>
      <a:spAutoFit/>
    </a:bodyPr>
    <a:lstStyle/>
    <a:p>
…
      <a:t>Sample Text</a:t>
…
    </a:p>
  </p:txBody>
</p:sp>
```

The resulting text that has now had the Inflate text warp applied to it is shown above. *end example*]

Using any of the presets listed under the ST_TextShapeType list below it is possible to apply a text warp to a run of DrawingML text via the following steps.

If you look at any of the text warps in the file format you notice that each consists of two paths. This corresponds to a top path (first one specified) and a bottom path (second one specified). Now the top path and the bottom path represent the top line and base line that the text needs to be warped to. This is done in the following way:

1. Compute the rectangle that the unwarped text resides in. (tightest possible rectangle around text, no white space except for “space characters”)
2. Take each of the quadratic and cubic Bezier curves that are used to calculate the original character and change their end points and control points by the following method…
3. Move a vertical line horizontally along the original text rectangle and find the horizontal percentage that a given end point or control point lives at. (.5 for the middle for instance)
4. Now do the same thing for this point vertically. Find the vertical percentage that this point lives at with the top and bottom of this text rectangle being the respective top and bottom bounds. (0.0 and 1.0 respectively)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

5. Now that we have the percentages for a given point in a Bezier equation we can map that to the new point in the warped text environment.
6. Going back to the top and bottom paths specified in the file format we can take these and flatten them out to a straight arc (top and bottom might be different lengths)
7. After they are straight we can measure them both horizontally to find the same percentage point that we found within the original text rectangle. (0.5 let's say)
8. So then we measure 50% along the top path and 50% along the bottom path, putting the paths back to their original curvy shapes.
9. Once we have these two points we can draw a line between them that serves as our vertical line in the original text rectangle [*Note*: This might not be truly vertical as 50% on the top does not always line up with 50% on the bottom. *end note*]
10. Taking this new line we then follow it from top to bottom the vertical percentage amount that we got from step 4.
11. This is then the new point that should be used in place of the old point in the original text rectangle.
12. We then continue doing these same steps for each of the end points and control points within the body of text. (is applied to a whole body of text only)

[*Note*: Horizxontal percentages begin at 0.0 and continue to 1.0, left to right. Vertical percentages begin at 0.0 and continue to 1.0, top to bottom. *end note*]

[*Note*: Since this is a shape it does have both a shape coordinate system and a path coordinate system. *end note*]

Attributes	Description

prst (Preset Warp Shape)	Specifies the preset geometry that is used for a shape warp on a piece of text. This preset can have any of the values in the enumerated list for ST_TextShapeType. This attribute is required in order for a text warp to be rendered.
	
	[*Example*: Consider the sample DrawingML below.
	
	`<p:sp>`
	`  <p:txBody>`
	`  <a:bodyPr wrap="none" rtlCol="0">`
	`  <a:prstTxWarp prst="textInflate">`
	`  </a:prstTxWarp>`
	`  <a:spAutoFit/>`
	`  </a:bodyPr>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  …`
	`  <a:t>Sample Text</a:t>`
	`  …`
	`  </a:p>`
	`  </p:txBody>`
	`</p:sp>`
	
	In the above example a preset text shape geometry has been used to define the warping

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

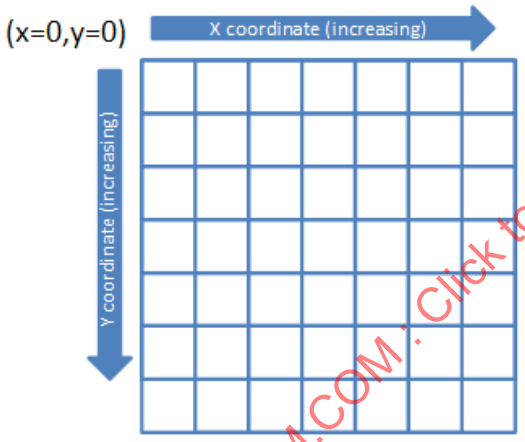
Attributes	Description
	shape. The shape utilized here is the sun shape. *end example*] The possible values for this attribute are defined by the ST_TextShapeType simple type (§20.1.10.76).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresetTextShape) is located in §A.4.1. *end note*]

20.1.9.20 pt (Shape Path Point)

This element specifies an x-y coordinate within the path coordinate space. This coordinate space is determined by the width and height attributes defined within the path element. A point is utilized by one of it's parent elements to specify the next point of interest in custom geometry shape. Depending on the parent element used the point can either have a line drawn to it or the cursor can simply be moved to this new location.

[*Note*: When specifying a point coordinate in path coordinate space it should be noted that the top left of the coordinate space is x=0, y=0 and the coordinate points for x grow to the right and for y grow down. This is illustrated in the diagram below.



end note]

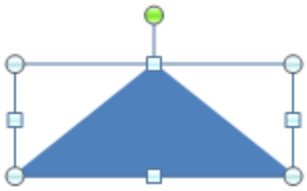
Specifies a position coordinate within the shape bounding box. It should be noted that this coordinate is placed within the shape bounding box using the transform coordinate system which is also called the shape coordinate system, as it encompasses the entire shape. The width and height for this coordinate system are specified within the ext transform element.

[*Example*: To highlight the differences in the coordinate systems consider the drawing of the following triangle. Notice that the dimensions of the triangle are specified using the shape coordinate system with EMUs as the units via the ext transform element. Thus we see this shape is 1705233 EMUs wide by 679622 EMUs tall. However when looking at how the path for this shape is drawn we see that the x and y values fall between 0 and

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

2. This is because the path coordinate system has the arbitrary dimensions of 2 for the width and 2 for the height. Thus we see that a y coordinate of 2 within the path coordinate system specifies a y coordinate of 679622 within the shape coordinate system for this particular case.



```
<a:xfrm>
  <a:off x="3200400" y="1600200"/>
  <a:ext cx="1705233" cy="679622"/>
</a:xfrm>
<a:custGeom>
  <a:avLst/>
    <a:gdLst/>
  <a:ahLst/>
  <a:cxnLst/>
  <a:rect l="0" t="0" r="0" b="0"/>
  <a:pathLst>
    <a:path w="2" h="2">
      <a:moveTo>
        <a:pt x="0" y="2"/>
      </a:moveTo>
      <a:lnTo>
        <a:pt x="2" y="2"/>
      </a:lnTo>
      <a:lnTo>
        <a:pt x="1" y="0"/>
      </a:lnTo>
      <a:close/>
    </a:path>
  </a:pathLst>
</a:custGeom>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

Attributes	Description
x (X-Coordinate)	Specifies the x coordinate for this position coordinate. The units for this coordinate space are defined by the width of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
y (Y-Coordinate)	Specifies the y coordinate for this position coordinate. The units for this coordinate space are defined by the height of the path coordinate system. This coordinate system is overlayed on top of the shape coordinate system thus occupying the entire shape bounding box. Because the units for within this coordinate space are determined by the path width and height an exact measurement unit cannot be specified here. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjPoint2D) is located in §A.4.1. *end note*]

20.1.9.21 quadBezTo (Draw Quadratic Bezier Curve To)

This element specifies to draw a quadratic bezier curve along the specified points. To specify a quadratic bezier curve there needs to be 2 points specified. The first is a control point used in the quadratic bezier calculation and the last is the ending point for the curve. The coordinate system used for this type of curve is the path coordinate system as this element is path specific.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Path2DQuadBezierTo) is located in §A.4.1. *end note*]

20.1.9.22 rect (Shape Text Rectangle)

This element specifies the rectangular bounding box for text within a custGeom shape. The default for this rectangle is the bounding box for the shape. This can be modified using this elements four attributes to inset or extend the text bounding box.

[*Note*: Text specified to reside within this shape text rectangle can flow outside this bounding box. Depending on the autofit options within the txBody element the text might not entirely reside within this shape text rectangle. *end note*]

Attributes	Description
b (Bottom Position)	Specifies the y coordinate of the bottom edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
l (Left)	Specifies the x coordinate of the left edge for a shape text rectangle. The units for this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
r (Right)	Specifies the x coordinate of the right edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).
t (Top)	Specifies the y coordinate of the top edge for a shape text rectangle. The units for this edge is specified in EMUs as the positioning here is based on the shape coordinate system. The width and height for this coordinate system are specified within the ext transform element. The possible values for this attribute are defined by the ST_AdjCoordinate simple type (§20.1.10.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GeomRect) is located in §A.4.1. *end note*]

20.1.10 Simple Types

This is the complete list of simple types dedicated to DrawingML framework.

20.1.10.1 ST_AdjAngle (Adjustable Angle Methods)

This simple type is an adjustable angle, either an absolute angle or a reference to a geometry guide. The units for an adjustable angle are 60,000ths of a degree.

This simple type is a union of the following types:

- The ST_Angle simple type (§20.1.10.3).
- The ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AdjAngle) is located in §A.4.1. *end note*]

20.1.10.2 ST_AdjCoordinate (Adjustable Coordinate Methods)

This simple type is an adjustable coordinate is either an absolute coordinate position or a reference to a geometry guide.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is a union of the following types:

- The ST_Coordinate simple type (§20.1.10.16).
- The ST_GeomGuideName simple type (§20.1.10.28).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AdjCoordinate) is located in §A.4.1. *end note*]

20.1.10.3 ST_Angle (Angle)

This simple type represents an angle in 60,000ths of a degree. Positive angles are clockwise (i.e., towards the positive y axis); negative angles are counter-clockwise (i.e., towards the negative y axis).

This simple type's contents are a restriction of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Angle) is located in §A.4.1. *end note*]

20.1.10.4 ST_AnimationBuildType (Animation Build Type)

This simple type specifies the ways that an animation can be built, or animated.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allAtOnce (Animate At Once)	Animate all objects as one.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationBuildType) is located in §A.4.1. *end note*]

20.1.10.5 ST_AnimationChartBuildType (Chart Animation Build Type)

This simple type specifies the ways that a chart animation can be built. That is, it specifies the way in which the objects within the chart should be animated.

This simple type is a union of the following types:

- The ST_AnimationBuildType simple type (§20.1.10.4).
- The ST_AnimationChartOnlyBuildType simple type (§20.1.10.6).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationChartBuildType) is located in §A.4.1. *end note*]

20.1.10.6 ST_AnimationChartOnlyBuildType (Chart only Animation Types)

This simple type specifies the build options available only for animating a chart. These options specify the manner in which the objects within the chart should be grouped and animated.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
category (Catefory)	Animate by each category
categoryEl (Category Element)	Animate by each element within the category
series (Series)	Animate by each series.
seriesEl (Series Element)	Animate by each element within the series

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationChartOnlyBuildType) is located in §A.4.1. *end note*]

20.1.10.7 ST_AnimationDgmBuildType (Diagram Animation Build Type)

This simple type specifies the ways that a diagram animation can be built. That is, it specifies the way in which the objects within the diagram graphical object should be animated.

This simple type is a union of the following types:

- The ST_AnimationBuildType simple type (§20.1.10.4).
- The ST_AnimationDgmOnlyBuildType simple type (§20.1.10.8).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationDgmBuildType) is located in §A.4.1. *end note*]

20.1.10.8 ST_AnimationDgmOnlyBuildType (Diagram only Animation Types)

This simple type specifies the build options available only for animating a diagram. These options specify the manner in which the objects within the chart should be grouped and animated.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lvlAtOnce (Each Level at Once)	Animate the diagram one level at a time, animating the whole level as one object
lvlOne (Level One-by-One)	Animate the diagram by the elements within a level, animating them one level element at a time.
one (Elements One-by-One)	Animate the diagram by elements. For a tree diagram the animation occurs by branch within the diagram tree.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimationDgmOnlyBuildType) is located in §A.4.1. *end note*]

20.1.10.9 ST_BevelPresetType (Bevel Presets)

Represents a preset for a type of bevel which can be applied to a shape in 3D. The bevel properties are applied differently depending on the type of bevel defined for a shape.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
angle (Angle)	[*Example*: Consider the following example of an `angle` bevel type applied to a shape: *end example*]
artDeco (Art Deco)	[*Example*: Consider the following example of an `artDeco` bevel type applied to a shape: *end example*]
circle (Circle)	[*Example*: Consider the following example of an `circle` bevel type applied to a shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
convex (Convex)	[*Example*: Consider the following example of an `convex` bevel type applied to a shape: *end example*]
coolSlant (Cool Slant)	[*Example*: Consider the following example of an `coolSlant` bevel type applied to a shape: *end example*]
cross (Cross)	[*Example*: Consider the following example of an `cross` bevel type applied to a shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	 end example]
divot (Divot)	[*Example*: Consider the following example of an `divot` bevel type applied to a shape: *end example*]
hardEdge (Hard Edge)	[*Example*: Consider the following example of an `hardEdge` bevel type applied to a shape: *end example*]
relaxedInset (Relaxed Inset)	[*Example*: Consider the following example of an

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	`relaxedInset` bevel type applied to a shape: *end example*]
riblet (Riblet)	[*Example*: Consider the following example of an `riblet` bevel type applied to a shape: *end example*]
slope (Slope)	[*Example*: Consider the following example of an `slope` bevel type applied to a shape: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
softRound (Soft Round)	[*Example*: Consider the following example of an `softRound` bevel type applied to a shape: *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BevelPresetType) is located in §A.4.1. *end note*]

20.1.10.10 ST_BlackWhiteMode (Black and White Mode)

This simple type specifies how an object should be rendered when specified to be in black and white mode.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Automatic)	Object rendered with automatic coloring
black (Black)	Object rendered with black-only coloring
blackGray (Black and Gray)	Object rendered with black and gray coloring
blackWhite (Black and White)	Object rendered within black and white coloring
clr (Color)	Object rendered with normal coloring
gray (Gray)	Object rendered with gray coloring
grayWhite (Gray and White)	Object rendered within gray and white coloring
hidden (Hidden)	Object rendered with hidden coloring
invGray (Inverse Gray)	Object rendered with inverse gray coloring
ltGray (Light Gray)	Object rendered with light gray coloring
white (White)	Object rendered within white coloirng

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlackWhiteMode) is located in §A.4.1. *end note*]

20.1.10.11 ST_BlendMode (Blend Mode)

This simple type describes how to render effects one on top of another.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
darken (Darken)	Darken
lighten (Lighten)	Lighten
mult (Multiply)	Multiply
over (Overlay)	Overlay
screen (Screen)	Screen

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlendMode) is located in §A.4.1. *end note*]

20.1.10.12 ST_BlipCompression (Blip Compression Type)

This type specifies the amount of compression that has been used for a particular binary large image or picture (blip).

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
email (Email Compression)	Compression size suitable for inclusion with email
hqprint (High Quality Printing Compression)	Compression size suitable for high quality printing
none (No Compression)	No compression was used
print (Printing Compression)	Compression size suitable for printing
screen (Screen Viewing Compression)	Compression size suitable for viewing on screen

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BlipCompression) is located in §A.4.1. *end note*]

20.1.10.13 ST_ChartBuildStep (Chart Animation Build Step)

This simple type specifies an animation build step within a chart animation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
allPts (All Points)	Animate all points within the chart for this animation build step
category (Category)	Animate a chart category for this animation build step
gridLegend (Grid and Legend)	Animate the chart grid and legend for this animation build step
ptInCategory (Category Points)	Animate a point in a chart category for this animation build step
ptInSeries (Series Points)	Animate a point in a chart series for this animation build step
series (Series)	Animate a chart series for this animation build step

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChartBuildStep) is located in §A.4.1. *end note*]

20.1.10.14 ST_ColorSchemeIndex (Theme Color Reference)

A reference to a color in the color scheme.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent 1)	Represents the accent 1 color.
accent2 (Accent 2)	Represents the accent 2 color.
accent3 (Accent 3)	Represents the accent 3 color.
accent4 (Accent 4)	Represents the accent 4 color.
accent5 (Accent 5)	Represents the accent 5 color.
accent6 (Accent 6)	Represents the accent 6 color.
dk1 (Dark 1)	Represents the first dark color.
dk2 (Dark 2)	Represents the second dark color.
folHlink (Followed Hyperlink)	Represents the followed hyperlink color.
hlink (Hyperlink)	Represents the hyperlink color.
lt1 (Light 1)	Represents the first light color.
lt2 (Light 2)	Represents the second light color.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ColorSchemeIndex) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.10.15 ST_CompoundLine (Compound Line Type)

This simple type specifies the compound line type that is to be used for lines with text such as underlines.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dbl (Double Lines)	Double lines of equal width
sng (Single Line)	Single line: one normal width
thickThin (Thick Thin Double Lines)	Double lines: one thick, one thin
thinThick (Thin Thick Double Lines)	Double lines: one thin, one thick
tri (Thin Thick Thin Triple Lines)	Three lines: thin, thick, thin

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CompoundLine) is located in §A.4.1. *end note*]

20.1.10.16 ST_Coordinate (Coordinate)

This simple type represents a one dimensional position or length as either:

- EMUs.
- A number followed immediately by a unit identifier.

This simple type is a union of the following types:

- The ST_CoordinateUnqualified simple type (§20.1.10.19).
- The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.1.10.17 ST_Coordinate32 (Coordinate Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing; its maximum size is 2147483647 EMUs.

Its contents can contain either:

- A whole number, whose contents consist of a measurement in EMUs (English Metric Units)
- A number immediately followed by a unit identifier

This simple type is a union of the following types:

- The ST_Coordinate32Unqualified simple type (§20.1.10.18).
- The ST_UniversalMeasure simple type (§22.9.2.15).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Coordinate32) is located in §A.4.1. *end note*]

20.1.10.18 ST_Coordinate32Unqualified (Coordinate Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing with the maximum size requirement being a 32 bit integer.

The units of measurement used here are EMUs (English Metric Units).

This simple type's contents are a restriction of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Coordinate32Unqualified) is located in §A.4.1. *end note*]

20.1.10.19 ST_CoordinateUnqualified (Coordinate)

This simple type represents a one dimensional position or length in EMUs.

This simple type's contents are a restriction of the W3C XML Schema long datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -27273042329600.
- This simple type has a maximum value of less than or equal to 27273042316900.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CoordinateUnqualified) is located in §A.4.1. *end note*]

20.1.10.20 ST_DgmBuildStep (Diagram Animation Build Steps)

This simple type specifies an animation build step within a diagram animation.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bg (Background)	Animate the diagram background for this animation build step
sp (Shape)	Animate a diagram shape for this animation build step

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DgmBuildStep) is located in §A.4.1. *end note*]

20.1.10.21 ST_DrawingElementId (Drawing Element ID)

This simple type specifies a unique integer identifier for each drawing element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DrawingElementId) is located in §A.4.1. *end note*]

20.1.10.22 ST_EffectContainerType (Effect Container Type)

This simple type determines the relationship between effects in a container, either sibling or tree.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
sib (Sibling)	Each effect is separately applied to the parent object. [*Example*: If the parent element contains an outer shadow and a reflection, the resulting effect is a shadow around the parent object and a reflection of the object. The reflection does not have a shadow. *end example*]
tree (Tree)	Each effect is applied to the result of the previous effect. [*Example*: If the parent element contains an outer shadow followed by a glow, the shadow is first applied to the parent object. Then, the glow is applied to the shadow (rather than the original object). The resulting effect would be a glowing shadow. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EffectContainerType) is located in §A.4.1. *end note*]

20.1.10.23 ST_FixedAngle (Fixed Angle)

This simple type represents a fixed range angle in 60000ths of a degree. Range from (-90, 90 degrees).

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than -5400000.
- This simple type has a maximum value of less than 5400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedAngle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.10.24 ST_FixedPercentage (Fixed Percentage)

This simple type represents a fixed percentage from negative one hundred to positive one hundred percent. See the union's member types for details.

This simple type is a union of the following types:

- The ST_FixedPercentage simple type (§22.9.2.3).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedPercentage) is located in §A.4.1. *end note*]

20.1.10.25 ST_FontCollectionIndex (Font Collection Index)

This simple type represents one of the fonts associated with the style.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
major (Major Font)	The major font of the style's font scheme.
minor (Minor Font)	The minor font of the style's font scheme.
none (None)	No font reference.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FontCollectionIndex) is located in §A.4.1. *end note*]

20.1.10.26 ST_FOVAngle (Field of View Angle)

Represents a positive angle in 60000ths of a degree. Range from [0, 180] degrees.

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 10800000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FOVAngle) is located in §A.4.1. *end note*]

20.1.10.27 ST_GeomGuideFormula (Geometry Guide Formula Properties)

This simple type specifies a geometry guide formula.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GeomGuideFormula) is located in §A.4.1. *end note*]

20.1.10.28 ST_GeomGuideName (Geometry Guide Name Properties)

This simple type specifies a geometry guide name.

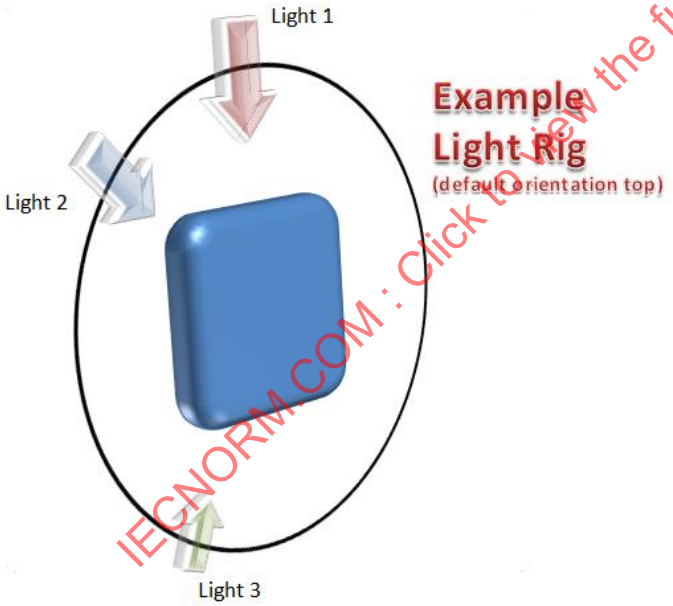
This simple type's contents are a restriction of the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GeomGuideName) is located in §A.4.1. *end note*]

20.1.10.29 ST_LightRigDirection (Light Rig Direction)

Represents the direction from which the light rig is positioned relative to the scene. The light rig, itself, can be made up of multiple lights in any orientation around a given shape. This simple type defines the orientation of the light rig as a whole, and not the individual lights within the rig. This means that because the direction of the light rig is left, that does not guarantee the light is coming from the left side of the shape, but rather the orientation of the rig as a whole is rotated to the left.

[*Example*: Consider the following example as a visual representation of a light rig oriented from the top of the shape in the center:



In this example we see that the light rig defines three lights (all in a single plane as represented by the black circular line). The lights defined in this representation can all have different intensities, which means, for this example, Light 3 and Light 2 look to have a more intense effect (or could even be a different color) than Light 1. One can imagine rotating this rig to the so that Light 1 is to the right of the shape when the light rig direction is defined to be right. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Three Point light rig type
- Circle bevel type
- Plastic material type
- Camera type defined by the `orthographicFront` preset
- Bevel width and height each equal to `190500`

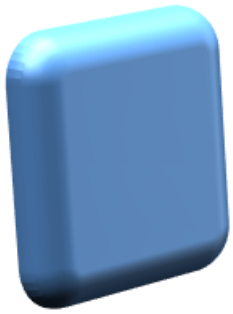
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	[*Example*: Consider the following example of a light direction from the bottom: *end example*]
bl (Bottom Left)	[*Example*: Consider the following example of a light direction from the bottom left: *end example*]
br (Bottom Right)	[*Example*: Consider the following example of a light direction from the bottom right:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
l (Left)	[*Example*: Consider the following example of a light direction from the left: *end example*]
r (Right)	[*Example*: Consider the following example of a light direction from the right:  *end example*]
t (Top)	[*Example*: Consider the following example of a light

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	direction from the top: *end example*]
tl (Top Left)	[*Example*: Consider the following example of a light direction from the top left: *end example*]
tr (Top Right)	[*Example*: Consider the following example of a light direction from the top right: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LightRigDirection) is located in §A.4.1. *end note*]

20.1.10.30 ST_LightRigType (Light Rig Type)

Represents a preset light right that can be applied to a shape. The light rig represents a group of lights oriented in a specific way relative to a 3D scene. The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Circle bevel type
- Warm Matte material type
- Camera type defined by the `perspectiveContrastingRightFacing` preset
- Bevel width and height each equal to `190500`

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
balanced (Light Rig Enum (Balanced))	Balanced
brightRoom (Bright Room)	[*Example*: Consider the following example of the `brightRoom` light rig applied to a basic shape: *end example*]
chilly (Chilly)	[*Example*: Consider the following example of the `chilly` light rig applied to a basic shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
contrasting (Contrasting)	[*Example*: Consider the following example of the `contrasting` light rig applied to a basic shape: *end example*]
flat (Flat)	[*Example*: Consider the following example of the `flat` light rig applied to a basic shape: *end example*]
flood (Flood)	[*Example*: Consider the following example of the `flood` light rig applied to a basic shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
freezing (Freezing)	[*Example*: Consider the following example of the `freezing` light rig applied to a basic shape: *end example*]
glow (Glow)	[*Example*: Consider the following example of the `glow` light rig applied to a basic shape: *end example*]
harsh (Harsh)	[*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

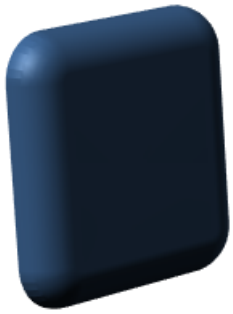
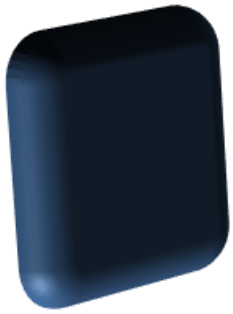
Enumeration Value	Description
	`harsh` light rig applied to a basic shape: *end example*]
legacyFlat1 (Legacy Flat 1)	[*Example*: Consider the following example of the `legacyFlat1` light rig applied to a basic shape: *end example*]
legacyFlat2 (Legacy Flat 2)	[*Example*: Consider the following example of the `legacyFlat2` light rig applied to a basic shape: *end example*]
legacyFlat3 (Legacy Flat 3)	[*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	`legacyFlat3` light rig applied to a basic shape: *end example*]
legacyFlat4 (Legacy Flat 4)	[*Example*: Consider the following example of the `legacyFlat4` light rig applied to a basic shape: *end example*]
legacyHarsh1 (Legacy Harsh 1)	[*Example*: Consider the following example of the `legacyHarsh1` light rig applied to a basic shape: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
legacyHarsh2 (Legacy Harsh 2)	[*Example*: Consider the following example of the `legacyHarsh2` light rig applied to a basic shape:  *end example*]
legacyHarsh3 (Legacy Harsh 3)	[*Example*: Consider the following example of the `legacyHarsh3` light rig applied to a basic shape: *end example*]
legacyHarsh4 (Legacy Harsh 4)	[*Example*: Consider the following example of the `legacyHarsh4` light rig applied to a basic shape:  *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
legacyNormal1 (Legacy Normal 1)	[*Example*: Consider the following example of the `legacyNormal1` light rig applied to a basic shape: *end example*]
legacyNormal2 (Legacy Normal 2)	[*Example*: Consider the following example of the `legacyNormal2` light rig applied to a basic shape: *end example*]
legacyNormal3 (Legacy Normal 3)	[*Example*: Consider the following example of the `legacyNormal3` light rig applied to a basic shape: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
legacyNormal4 (Legacy Normal 4)	[*Example*: Consider the following example of the `legacyNormal4` light rig applied to a basic shape: *end example*]
morning (Morning)	[*Example*: Consider the following example of the `morning` light rig applied to a basic shape: *end example*]
soft (Soft)	[*Example*: Consider the following example of the `soft` light rig applied to a basic shape: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
sunrise (Sunrise)	[*Example*: Consider the following example of the `sunrise` light rig applied to a basic shape: *end example*]
sunset (Sunset)	[*Example*: Consider the following example of the `sunset` light rig applied to a basic shape: *end example*]
threePt (Three Point)	[*Example*: Consider the following example of the `threePt` light rig applied to a basic shape: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
twoPt (Two Point)	[*Example*: Consider the following example of the `twoPt` light rig applied to a basic shape: *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LightRigType) is located in §A.4.1. *end note*]

20.1.10.31 ST_LineCap (End Line Cap)

This simple type specifies how to cap the ends of lines. This also affects the ends of line segments for dashed lines.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
flat (Flat Line Cap)	Line ends at end point.
rnd (Round Line Cap)	Rounded ends. Semi-circle protrudes by half line width.
sq (Square Line Cap)	Square protrudes by half line width.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineCap) is located in §A.4.1. *end note*]

20.1.10.32 ST_LineEndLength (Line End Length)

This simple type represents the length of the line end decoration (e.g., arrowhead) relative to the width of the line itself.

[*Example:* See the example images below. These samples have an arrow line end type and medium line end width. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lg (Large)	Large
med (Medium)	Medium
sm (Small)	Small

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndLength) is located in §A.4.1. *end note*]

20.1.10.33 ST_LineEndType (Line End Type)

This simple type represents the shape decoration that appears at the ends of lines. For example, one choice is an arrow head.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
arrow (Arrow Head)	Line arrow head
diamond (Diamond)	Diamond
none (None)	No end
oval (Oval)	Oval
stealth (Stealth Arrow)	Stealth arrow head
triangle (Triangle Arrow Head)	Triangle arrow head

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndType) is located in §A.4.1. *end note*]

20.1.10.34 ST_LineEndWidth (Line End Width)

This simple type represents the width of the line end decoration (e.g., arrowhead) relative to the width of the line itself.

[*Example:* See the example images below. These samples have an arrow line end type and medium line end length. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
lg (Large)	→ Large
med (Medium)	→ Medium
sm (Small)	→ Small

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineEndWidth) is located in §A.4.1. *end note*]

20.1.10.35 ST_LineWidth (Line Width)

This simple type specifies the width of a line in EMUs. 1 pt = 12700 EMUs.

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 20116800.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LineWidth) is located in §A.4.1. *end note*]

20.1.10.36 ST_OnOffStyleType (On/Off Style Type)

This simple type represents whether a style property should be applied.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
def (Default)	Follow parent settings. For a themed property, follow the theme settings. For an unthemed property, follow the parent setting in the property inheritance chain.
off (Off)	Property is off.
on (On)	Property is on.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OnOffStyleType) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.10.37 ST_PathFillMode (Path Fill Mode)

This simple type specifies the manner in which a path should be filled. The lightening and darkening of a path allow for certain parts of the shape to be colored lighter of darker depending on user preference.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
darken (Darken Path Fill)	This specifies that the corresponding path should have a darker shaded color applied to it's fill.
darkenLess (Darken Path Fill Less)	This specifies that the corresponding path should have a slightly darker shaded color applied to it's fill.
lighten (Lighten Path Fill)	This specifies that the corresponding path should have a lightly shaded color applied to it's fill.
lightenLess (Lighten Path Fill Less)	This specifies that the corresponding path should have a slightly lighter shaded color applied to it's fill.
none (No Path Fill)	This specifies that the corresponding path should have no fill.
norm (Normal Path Fill)	This specifies that the corresponding path should have a normally shaded color applied to it's fill.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PathFillMode) is located in §A.4.1. *end note*]

20.1.10.38 ST_PathShadeType (Path Shade Type)

This simple type describes the shape of path to follow for a path gradient shade.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
circle (Circle)	Gradient follows a circular path
rect (Rectangle)	Gradient follows a rectangular path
shape (Shape)	Gradient follows the shape

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PathShadeType) is located in §A.4.1. *end note*]

20.1.10.39 ST_PenAlignment (Alignment Type)

This simple type specifies the Pen Alignment type for use within a text body.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center Alignment)	Center pen (line drawn at center of path stroke).
in (Inset Alignment)	Inset pen (the pen is aligned on the inside of the edge of the path).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PenAlignment) is located in §A.4.1. *end note*]

20.1.10.40 ST_Percentage (Percentage)

This simple type specifies that its contents will contain a percentage value. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Percentage) is located in §A.4.1. *end note*]

20.1.10.41 ST_PitchFamily (Pitch Family)

This simple type specifies a font pitch.

[*Note:* Although the type name is ST_PitchFamily, the integer value of this attribute specifies the font family with the higher 4 bits and the font pitch with the lower 4 bits. *end note*]

This simple type's contents are a restriction of the W3C XML Schema byte datatype.

This simple type is restricted to the values listed in the following table:

Value	Description
0x00	DEFAULT PITCH + UNKNOWN FONT FAMILY
0x01	FIXED PITCH + UNKNOWN FONT FAMILY
0x02	VARIABLE PITCH + UNKNOWN FONT FAMILY
0x10	DEFAULT PITCH + ROMAN FONT FAMILY
0x11	FIXED PITCH + ROMAN FONT FAMILY
0x12	VARIABLE PITCH + ROMAN FONT FAMILY
0x20	DEFAULT PITCH + SWISS FONT FAMILY
0x21	FIXED PITCH + SWISS FONT FAMILY
0x22	VARIABLE PITCH + SWISS FONT FAMILY

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Value	Description
0x30	DEFAULT PITCH + MODERN FONT FAMILY
0x31	FIXED PITCH + MODERN FONT FAMILY
0x32	VARIABLE PITCH + MODERN FONT FAMILY
0x40	DEFAULT PITCH + SCRIPT FONT FAMILY
0x41	FIXED PITCH + SCRIPT FONT FAMILY
0x42	VARIABLE PITCH + SCRIPT FONT FAMILY
0x50	DEFAULT PITCH + DECORATIVE FONT FAMILY
0x51	FIXED PITCH + DECORATIVE FONT FAMILY
0x52	VARIABLE PITCH + DECORATIVE FONT FAMILY

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PitchFamily) is located in §A.3. *end note*]

20.1.10.42 ST_PositiveCoordinate (Positive Coordinate)

This simple type represents a positive position or length in EMUs.

This simple type's contents are a restriction of the W3C XML Schema long datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 27273042316900.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveCoordinate) is located in §A.4.1. *end note*]

20.1.10.43 ST_PositiveCoordinate32 (Positive Coordinate Point)

This simple type specifies the a positive coordinate point that has a maximum size of 32 bits.

The units of measurement used here are EMUs (English Metric Units).

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveCoordinate32) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.1.10.44 ST_PositiveFixedAngle (Positive Fixed Angle)

This simple type represents a positive angle in 60000ths of a degree. Range from [0, 360 degrees).

This simple type's contents are a restriction of the ST_Angle datatype (§20.1.10.3).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than 21600000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedAngle) is located in §A.4.1. *end note*]

20.1.10.45 ST_PositiveFixedPercentage (Positive Fixed Percentage)

This simple type specifies that its contents will contain a positive percentage value from zero through one hundred percent. See the union's member types for details.

This simple type is a union of the following types:

- The ST_PositiveFixedPercentage simple type (§22.9.2.10).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedPercentage) is located in §A.4.1. *end note*]

20.1.10.46 ST_PositivePercentage (Positive Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage value. See the union's member types for details.

This simple type is a union of the following types:

- The ST_PositivePercentage simple type (§22.9.2.11).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositivePercentage) is located in §A.4.1. *end note*]

20.1.10.47 ST_PresetCameraType (Preset Camera Type)

These enumeration values represent different algorithmic methods for setting all camera properties, including position. The following example images below are all based off the following shape:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this image, we can see the shape has a camera pointing directly at the front face.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
isometricBottomDown (Isometric Bottom Down)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricBottomUp (Isometric Bottom Up)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricLeftDown (Isometric Left Down)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
isometricLeftUp (Isometric Left Up)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis1Left (Isometric Off Axis 1 Left)	[*Example*: Consider the following example of the camera preset type: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
isometricOffAxis1Right (Isometric Off Axis 1 Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis1Top (Isometric Off Axis 1 Top)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis2Left (Isometric Off Axis 2 Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis2Right (Isometric Off Axis 2 Right)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
isometricOffAxis2Top (Isometric Off Axis 2 Top)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis3Bottom (Isometric Off Axis 3 Bottom)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis3Left (Isometric Off Axis 3 Left)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
isometricOffAxis3Right (Isometric Off Axis 3 Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis4Bottom (Isometric Off Axis 4 Bottom)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricOffAxis4Left (Isometric Off Axis 4 Left)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
isometricOffAxis4Right (Isometric Off Axis 4 Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricRightDown (Isometric Right Down)	[*Example*: Consider the following example of the camera preset type: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
isometricRightUp (Isometric Right Up)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricTopDown (Isometric Top Down)	[*Example*: Consider the following example of the camera preset type: *end example*]
isometricTopUp (Isometric Top Up)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueBottom (Legacy Oblique Bottom)	[*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	camera preset type: *end example*]
legacyObliqueBottomLeft (Legacy Oblique Bottom Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueBottomRight (Legacy Oblique Bottom Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueFront (Legacy Oblique Front)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
legacyObliqueLeft (Legacy Oblique Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueRight (Legacy Oblique Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueTop (Legacy Oblique Top)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
legacyObliqueTopLeft (Legacy Oblique Top Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyObliqueTopRight (Legacy Oblique Top Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveBottom (Legacy Perspective Bottom)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
legacyPerspectiveBottomLeft (Legacy Perspective Bottom Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveBottomRight (Legacy Perspective Bottom Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveFront (Legacy Perspective Front)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
legacyPerspectiveLeft (Legacy Perspective Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveRight (Legacy Perspective Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveTop (Legacy Perspective Top)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
legacyPerspectiveTopLeft (Legacy Perspective Top Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
legacyPerspectiveTopRight (Legacy Perspective Top Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
obliqueBottom (Oblique Bottom)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
obliqueBottomLeft (Oblique Bottom Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
obliqueBottomRight (Oblique Bottom Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
obliqueLeft (Oblique Left)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
obliqueRight (Oblique Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
obliqueTop (Oblique Top)	[*Example*: Consider the following example of the camera preset type: *end example*]
obliqueTopLeft (Oblique Top Left)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
obliqueTopRight (Oblique Top Right)	[*Example*: Consider the following example of the camera preset type: *end example*]
orthographicFront (Orthographic Front)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveAbove (Orthographic Above)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
perspectiveAboveLeftFacing (Perspective Above Left Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveAboveRightFacing (Perspective Above Right Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveBelow (Perspective Below)	[*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	camera preset type: *end example*]
perspectiveContrastingLeftFacing (Perspective Contrasting Left Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveContrastingRightFacing (Perspective Contrasting Right Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveFront (Perspective Front)	[*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	camera preset type: *end example*]
perspectiveHeroicExtremeLeftFacing (Perspective Heroic Extreme Left Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveHeroicExtremeRightFacing (Perspective Heroic Extreme Right Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveHeroicLeftFacing (Perspective Heroic Left Facing)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	 end example]
perspectiveHeroicRightFacing (Perspective Heroic Right Facing)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveLeft (Perspective Left)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveRelaxed (Perspective Relaxed)	[*Example*: Consider the following example of the camera preset type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
perspectiveRelaxedModerately (Perspective Relaxed Moderately)	[*Example*: Consider the following example of the camera preset type: *end example*]
perspectiveRight (Perspective Right)	[*Example*: Consider the following example of the camera preset type: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetCameraType) is located in §A.4.1. *end note*]

20.1.10.48 ST_PresetColorVal (Preset Color Value)

This simple type represents a preset color value.

©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aliceBlue (Alice Blue Preset Color)	Specifies a color with RGB value (240,248,255)
antiqueWhite (Antique White Preset Color)	Specifies a color with RGB value (250,235,215)
aqua (Aqua Preset Color)	Specifies a color with RGB value (0,255,255)
aquamarine (Aquamarine Preset Color)	Specifies a color with RGB value (127,255,212)
azure (Azure Preset Color)	Specifies a color with RGB value (240,255,255)
beige (Beige Preset Color)	Specifies a color with RGB value (245,245,220)
bisque (Bisque Preset Color)	Specifies a color with RGB value (255,228,196)
black (Black Preset Color)	Specifies a color with RGB value (0,0,0)
blanchedAlmond (Blanched Almond Preset Color)	Specifies a color with RGB value (255,235,205)
blue (Blue Preset Color)	Specifies a color with RGB value (0,0,255)
blueViolet (Blue Violet Preset Color)	Specifies a color with RGB value (138,43,226)
brown (Brown Preset Color)	Specifies a color with RGB value (165,42,42)
burlyWood (Burly Wood Preset Color)	Specifies a color with RGB value (222,184,135)
cadetBlue (Cadet Blue Preset Color)	Specifies a color with RGB value (95,158,160)
chartreuse (Chartreuse Preset Color)	Specifies a color with RGB value (127,255,0)
chocolate (Chocolate Preset Color)	Specifies a color with RGB value (210,105,30)
coral (Coral Preset Color)	Specifies a color with RGB value (255,127,80)
cornflowerBlue (Cornflower Blue Preset Color)	Specifies a color with RGB value (100,149,237)
cornsilk (Cornsilk Preset Color)	Specifies a color with RGB value (255,248,220)
crimson (Crimson Preset Color)	Specifies a color with RGB value (220,20,60)
cyan (Cyan Preset Color)	Specifies a color with RGB value (0,255,255)
darkBlue (Dark Blue Preset Color)	Specifies a color with RGB value (0,0,139)
darkCyan (Dark Cyan Preset Color)	Specifies a color with RGB value (0,139,139)
darkGoldenrod (Dark Goldenrod Preset Color)	Specifies a color with RGB value (184,134,11)
darkGray (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
darkGreen (Dark Green Preset Color)	Specifies a color with RGB value (0,100,0)
darkGrey (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
darkKhaki (Dark Khaki Preset Color)	Specifies a color with RGB value (189,183,107)
darkMagenta (Dark Magenta Preset Color)	Specifies a color with RGB value (139,0,139)
darkOliveGreen (Dark Olive Green Preset Color)	Specifies a color with RGB value (85,107,47)
darkOrange (Dark Orange Preset Color)	Specifies a color with RGB value (255,140,0)
darkOrchid (Dark Orchid Preset Color)	Specifies a color with RGB value (153,50,204)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
darkRed (Dark Red Preset Color)	Specifies a color with RGB value (139,0,0)
darkSalmon (Dark Salmon Preset Color)	Specifies a color with RGB value (233,150,122)
darkSeaGreen (Dark Sea Green Preset Color)	Specifies a color with RGB value (143,188,143)
darkSlateBlue (Dark Slate Blue Preset Color)	Specifies a color with RGB value (72,61,139)
darkSlateGray (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
darkSlateGrey (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
darkTurquoise (Dark Turquoise Preset Color)	Specifies a color with RGB value (0,206,209)
darkViolet (Dark Violet Preset Color)	Specifies a color with RGB value (148,0,211)
deepPink (Deep Pink Preset Color)	Specifies a color with RGB value (255,20,147)
deepSkyBlue (Deep Sky Blue Preset Color)	Specifies a color with RGB value (0,191,255)
dimGray (Dim Gray Preset Color)	Specifies a color with RGB value (105,105,105)
dimGrey (Dim Gray Preset Color)	Specifies a color with RGB value (105,105,105)
dkBlue (Dark Blue Preset Color)	Specifies a color with RGB value (0,0,139)
dkCyan (Dark Cyan Preset Color)	Specifies a color with RGB value (0,139,139)
dkGoldenrod (Dark Goldenrod Preset Color)	Specifies a color with RGB value (184,134,11)
dkGray (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
dkGreen (Dark Green Preset Color)	Specifies a color with RGB value (0,100,0)
dkGrey (Dark Gray Preset Color)	Specifies a color with RGB value (169,169,169)
dkKhaki (Dark Khaki Preset Color)	Specifies a color with RGB value (189,183,107)
dkMagenta (Dark Magenta Preset Color)	Specifies a color with RGB value (139,0,139)
dkOliveGreen (Dark Olive Green Preset Color)	Specifies a color with RGB value (85,107,47)
dkOrange (Dark Orange Preset Color)	Specifies a color with RGB value (255,140,0)
dkOrchid (Dark Orchid Preset Color)	Specifies a color with RGB value (153,50,204)
dkRed (Dark Red Preset Color)	Specifies a color with RGB value (139,0,0)
dkSalmon (Dark Salmon Preset Color)	Specifies a color with RGB value (233,150,122)
dkSeaGreen (Dark Sea Green Preset Color)	Specifies a color with RGB value (143,188,139)
dkSlateBlue (Dark Slate Blue Preset Color)	Specifies a color with RGB value (72,61,139)
dkSlateGray (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
dkSlateGrey (Dark Slate Gray Preset Color)	Specifies a color with RGB value (47,79,79)
dkTurquoise (Dark Turquoise Preset Color)	Specifies a color with RGB value (0,206,209)
dkViolet (Dark Violet Preset Color)	Specifies a color with RGB value (148,0,211)
dodgerBlue (Dodger Blue Preset Color)	Specifies a color with RGB value (30,144,255)
firebrick (Firebrick Preset Color)	Specifies a color with RGB value (178,34,34)
floralWhite (Floral White Preset Color)	Specifies a color with RGB value (255,250,240)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
forestGreen (Forest Green Preset Color)	Specifies a color with RGB value (34,139,34)
fuchsia (Fuchsia Preset Color)	Specifies a color with RGB value (255,0,255)
gainsboro (Gainsboro Preset Color)	Specifies a color with RGB value (220,220,220)
ghostWhite (Ghost White Preset Color)	Specifies a color with RGB value (248,248,255)
gold (Gold Preset Color)	Specifies a color with RGB value (255,215,0)
goldenrod (Goldenrod Preset Color)	Specifies a color with RGB value (218,165,32)
gray (Gray Preset Color)	Specifies a color with RGB value (128,128,128)
green (Green Preset Color)	Specifies a color with RGB value (0,128,0)
greenYellow (Green Yellow Preset Color)	Specifies a color with RGB value (173,255,47)
grey (Gray Preset Color)	Specifies a color with RGB value (128,128,128)
honeydew (Honeydew Preset Color)	Specifies a color with RGB value (240,255,240)
hotPink (Hot Pink Preset Color)	Specifies a color with RGB value (255,105,180)
indianRed (Indian Red Preset Color)	Specifies a color with RGB value (205,92,92)
indigo (Indigo Preset Color)	Specifies a color with RGB value (75,0,130)
ivory (Ivory Preset Color)	Specifies a color with RGB value (255,255,240)
khaki (Khaki Preset Color)	Specifies a color with RGB value (240,230,140)
lavender (Lavender Preset Color)	Specifies a color with RGB value (230,230,250)
lavenderBlush (Lavender Blush Preset Color)	Specifies a color with RGB value (255,240,245)
lawnGreen (Lawn Green Preset Color)	Specifies a color with RGB value (124,252,0)
lemonChiffon (Lemon Chiffon Preset Color)	Specifies a color with RGB value (255,250,205)
lightBlue (Light Blue Preset Color)	Specifies a color with RGB value (173,216,230)
lightCoral (Light Coral Preset Color)	Specifies a color with RGB value (240,128,128)
lightCyan (Light Cyan Preset Color)	Specifies a color with RGB value (224,255,255)
lightGoldenrodYellow (Light Goldenrod Yellow Preset Color)	Specifies a color with RGB value (250,250,210)
lightGray (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
lightGreen (Light Green Preset Color)	Specifies a color with RGB value (144,238,144)
lightGrey (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
lightPink (Light Pink Preset Color)	Specifies a color with RGB value (255,182,193)
lightSalmon (Light Salmon Preset Color)	Specifies a color with RGB value (255,160,122)
lightSeaGreen (Light Sea Green Preset Color)	Specifies a color with RGB value (32,178,170)
lightSkyBlue (Light Sky Blue Preset Color)	Specifies a color with RGB value (135,206,250)
lightSlateGray (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
lightSlateGrey (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
lightSteelBlue (Light Steel Blue Preset Color)	Specifies a color with RGB value (176,196,222)
lightYellow (Light Yellow Preset Color)	Specifies a color with RGB value (255,255,224)
lime (Lime Preset Color)	Specifies a color with RGB value (0,255,0)
limeGreen (Lime Green Preset Color)	Specifies a color with RGB value (50,205,50)
linen (Linen Preset Color)	Specifies a color with RGB value (250,240,230)
ltBlue (Light Blue Preset Color)	Specifies a color with RGB value (173,216,230)
ltCoral (Light Coral Preset Color)	Specifies a color with RGB value (240,128,128)
ltCyan (Light Cyan Preset Color)	Specifies a color with RGB value (224,255,255)
ltGoldenrodYellow (Light Goldenrod Yellow Preset Color)	Specifies a color with RGB value (250,250,120)
ltGray (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
ltGreen (Light Green Preset Color)	Specifies a color with RGB value (144,238,144)
ltGrey (Light Gray Preset Color)	Specifies a color with RGB value (211,211,211)
ltPink (Light Pink Preset Color)	Specifies a color with RGB value (255,182,193)
ltSalmon (Light Salmon Preset Color)	Specifies a color with RGB value (255,160,122)
ltSeaGreen (Light Sea Green Preset Color)	Specifies a color with RGB value (32,178,170)
ltSkyBlue (Light Sky Blue Preset Color)	Specifies a color with RGB value (135,206,250)
ltSlateGray (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
ltSlateGrey (Light Slate Gray Preset Color)	Specifies a color with RGB value (119,136,153)
ltSteelBlue (Light Steel Blue Preset Color)	Specifies a color with RGB value (176,196,222)
ltYellow (Light Yellow Preset Color)	Specifies a color with RGB value (255,255,224)
magenta (Magenta Preset Color)	Specifies a color with RGB value (255,0,255)
maroon (Maroon Preset Color)	Specifies a color with RGB value (128,0,0)
medAquamarine (Medium Aquamarine Preset Color)	Specifies a color with RGB value (102,205,170)
medBlue (Medium Blue Preset Color)	Specifies a color with RGB value (0,0,205)
mediumAquamarine (Medium Aquamarine Preset Color)	Specifies a color with RGB value (102,205,170)
mediumBlue (Medium Blue Preset Color)	Specifies a color with RGB value (0,0,205)
mediumOrchid (Medium Orchid Preset Color)	Specifies a color with RGB value (186,85,211)
mediumPurple (Medium Purple Preset Color)	Specifies a color with RGB value (147,112,219)
mediumSeaGreen (Medium Sea Green Preset Color)	Specifies a color with RGB value (60,179,113)
mediumSlateBlue (Medium Slate Blue Preset Color)	Specifies a color with RGB value (123,104,238)
mediumSpringGreen (Medium Spring Green Preset Color)	Specifies a color with RGB value (0,250,154)
mediumTurquoise (Medium Turquoise Preset Color)	Specifies a color with RGB value (72,209,204)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
mediumVioletRed (Medium Violet Red Preset Color)	Specifies a color with RGB value (199,21,133)
medOrchid (Medium Orchid Preset Color)	Specifies a color with RGB value (186,85,211)
medPurple (Medium Purple Preset Color)	Specifies a color with RGB value (147,112,219)
medSeaGreen (Medium Sea Green Preset Color)	Specifies a color with RGB value (60,179,113)
medSlateBlue (Medium Slate Blue Preset Color)	Specifies a color with RGB value (123,104,238)
medSpringGreen (Medium Spring Green Preset Color)	Specifies a color with RGB value (0,250,154)
medTurquoise (Medium Turquoise Preset Color)	Specifies a color with RGB value (72,209,204)
medVioletRed (Medium Violet Red Preset Color)	Specifies a color with RGB value (199,21,133)
midnightBlue (Midnight Blue Preset Color)	Specifies a color with RGB value (25,25,112)
mintCream (Mint Cream Preset Color)	Specifies a color with RGB value (245,255,250)
mistyRose (Misty Rose Preset Color)	Specifies a color with RGB value (255,228,225)
moccasin (Moccasin Preset Color)	Specifies a color with RGB value (255,228,181)
navajoWhite (Navajo White Preset Color)	Specifies a color with RGB value (255,222,173)
navy (Navy Preset Color)	Specifies a color with RGB value (0,0,128)
oldLace (Old Lace Preset Color)	Specifies a color with RGB value (253,245,230)
olive (Olive Preset Color)	Specifies a color with RGB value (128,128,0)
oliveDrab (Olive Drab Preset Color)	Specifies a color with RGB value (107,142,35)
orange (Orange Preset Color)	Specifies a color with RGB value (255,165,0)
orangeRed (Orange Red Preset Color)	Specifies a color with RGB value (255,69,0)
orchid (Orchid Preset Color)	Specifies a color with RGB value (218,112,214)
paleGoldenrod (Pale Goldenrod Preset Color)	Specifies a color with RGB value (238,232,170)
paleGreen (Pale Green Preset Color)	Specifies a color with RGB value (152,251,152)
paleTurquoise (Pale Turquoise Preset Color)	Specifies a color with RGB value (175,238,238)
paleVioletRed (Pale Violet Red Preset Color)	Specifies a color with RGB value (219,112,147)
papayaWhip (Papaya Whip Preset Color)	Specifies a color with RGB value (255,239,213)
peachPuff (Peach Puff Preset Color)	Specifies a color with RGB value (255,218,185)
peru (Peru Preset Color)	Specifies a color with RGB value (205,133,63)
pink (Pink Preset Color)	Specifies a color with RGB value (255,192,203)
plum (Plum Preset Color)	Specifies a color with RGB value (221,160,221)
powderBlue (Powder Blue Preset Color)	Specifies a color with RGB value (176,224,230)
purple (Purple Preset Color)	Specifies a color with RGB value (128,0,128)
red (Red Preset Color)	Specifies a color with RGB value (255,0,0)
rosyBrown (Rosy Brown Preset Color)	Specifies a color with RGB value (188,143,143)
royalBlue (Royal Blue Preset Color)	Specifies a color with RGB value (65,105,225)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
saddleBrown (Saddle Brown Preset Color)	Specifies a color with RGB value (139,69,19)
salmon (Salmon Preset Color)	Specifies a color with RGB value (250,128,114)
sandyBrown (Sandy Brown Preset Color)	Specifies a color with RGB value (244,164,96)
seaGreen (Sea Green Preset Color)	Specifies a color with RGB value (46,139,87)
seaShell (Sea Shell Preset Color)	Specifies a color with RGB value (255,245,238)
sienna (Sienna Preset Color)	Specifies a color with RGB value (160,82,45)
silver (Silver Preset Color)	Specifies a color with RGB value (192,192,192)
skyBlue (Sky Blue Preset Color)	Specifies a color with RGB value (135,206,235)
slateBlue (Slate Blue Preset Color)	Specifies a color with RGB value (106,90,205)
slateGray (Slate Gray Preset Color)	Specifies a color with RGB value (112,128,144)
slateGrey (Slate Gray Preset Color)	Specifies a color with RGB value (112,128,144)
snow (Snow Preset Color)	Specifies a color with RGB value (255,250,250)
springGreen (Spring Green Preset Color)	Specifies a color with RGB value (0,255,127)
steelBlue (Steel Blue Preset Color)	Specifies a color with RGB value (70,130,180)
tan (Tan Preset Color)	Specifies a color with RGB value (210,180,140)
teal (Teal Preset Color)	Specifies a color with RGB value (0,128,128)
thistle (Thistle Preset Color)	Specifies a color with RGB value (216,191,216)
tomato (Tomato Preset Color)	Specifies a color with RGB value (255,99,71)
turquoise (Turquoise Preset Color)	Specifies a color with RGB value (64,224,208)
violet (Violet Preset Color)	Specifies a color with RGB value (238,130,238)
wheat (Wheat Preset Color)	Specifies a color with RGB value (245,222,179)
white (White Preset Color)	Specifies a color with RGB value (255,255,255)
whiteSmoke (White Smoke Preset Color)	Specifies a color with RGB value (245,245,245)
yellow (Yellow Preset Color)	Specifies a color with RGB value (255,255,0)
yellowGreen (Yellow Green Preset Color)	Specifies a color with RGB value (154,205,50)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetColorVal) is located in §A.4.1. *end note*]

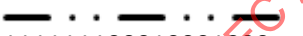
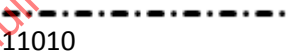
20.1.10.49 ST_PresetLineDashVal (Preset Line Dash Value)

This simple type represents preset line dash values. The description for each style shows an illustration of the line style. Each style also contains a precise binary representation of the repeating dash style. Each 1 corresponds to a line segment of the same length as the line width, and each 0 corresponds to a space of the same length as the line width.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dash (Dash)	1111000
dashDot (Dash Dot)	11110001000
dot (Dot)	1000
lgDash (Large Dash)	11111111000
lgDashDot (Large Dash Dot)	111111110001000
lgDashDotDot (Large Dash Dot Dot)	 1111111100010001000
solid (Solid)	1
sysDash (System Dash)	1110
sysDashDot (System Dash Dot)	 111010
sysDashDotDot (System Dash Dot Dot)	11101010
sysDot (System Dot)	10

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetLineDashVal) is located in §A.4.1. *end note*]

20.1.10.50 ST_PresetMaterialType (Preset Material Type)

Describes surface appearance of a shape. The material type combines with lighting characteristics to create the final look and feel of a shape. The set of material properties which can be combined together to create the presets below consist of the following characteristics:

- Specular color – This defines the color of the highlight associated with the material.
- Specular power – This defines the size and how intense the highlight is. Smaller values provide a larger, but less intense highlight, while larger values provide a smaller, but more intense highlight.
- Diffuse color – This defines the perceived color of the material where an object is directly illuminated by a light source. Generally speaking, the default color here would be based on the shape fill color.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Ambient color – This defines the perceived color of the material where an object is not directly illuminated by a light source. Generally speaking, the default color here would be based on the shape fill color.
- Emissive color – This defines the color of a light which is perceived to be given off by an object.
- Diffuse Fresnel effect – This is an effect that either darkens (approaches black) or lightens (approaches white) the diffuse color of the material type at glancing angles. Positive values cause the material to become brighter, negative values cause the material to become darker.
- Alpha Fresnel effect – This is an effect that either makes the material more opaque or more transparent at glancing angles. Positive values cause the material to become more opaque, negative values cause the material to become more transparent.

In the following examples, the exact values given for certain properties should be understood to be relative values in order to provide a reference. These values could be different depending upon technologies used to render the material types. The following properties were used to define the shape used in the image examples below:

- Rounded rectangle shape
- Circle bevel type
- Three Point light rig type
- Camera type defined by the `perspectiveContrastingRightFacing` preset
- Bevel width and height each equal to `190500`

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clear (Clear)	The `clear` material type has the following characteristics: • Specular Color: light gray • Diffuse Color: shape fill color with 90% alpha • Ambient Color: shape fill color with 90% alpha • Emissive Color: black • Diffuse Fresnel value: -8 • Alpha Fresnel value: 1 [*Example*: Consider the following example of the `clear` material type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 end example]
dkEdge (Dark Edge)	The dkEdge material type has the following characteristics: • Specular Color: white • Specular Power value: 35 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: -2 [*Example*: Consider the following example of the dkEdge material type: *end example*]
flat (Flat)	The flat material type has the following characteristics: • Specular Color: very light gray • Specular Power value: 50 • Diffuse Color: black • Ambient Color: black • Emissive Color: shape fill color • Diffuse Fresnel value: -4

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: Consider the following example of the `flat` material type: *end example*]
legacyMatte (Legacy Matte)	The `legacyMatte` material type has the following characteristics: • Specular Color: black • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: -4 [*Example*: Consider the following example of the `legacyMatte` material type: *end example*]
legacyMetal (Legacy Metal)	The `legacyMetal` material type has the following characteristics: • Specular Color: shape fill color • Specular Power value: 32 • Diffuse Color: shape fill color darkened by adding black • Ambient Color: shape fill color darkened by adding black

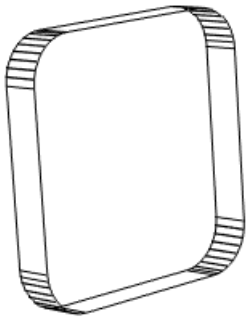
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	• Emissive Color: black [*Example*: Consider the following example of the `legacyMetal` material type: *end example*]
legacyPlastic (Legacy Plastic)	The `legacyPlastic` material type has the following characteristics: • Specular Color: white • Specular Power value: 32 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `legacyPlastic` material type: *end example*]
legacyWireframe (Legacy Wireframe)	The `legacyWireframe` material type has none of the associated material properties and is based on a wireframe interpretation of the shape. [*Example*: Consider the following example of the `legacyWireframe` material type:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
matte (Matte)	The `matte` material type has the following characteristics: • Specular Color: black • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `matte` material type: *end example*]
metal (Metal)	The `metal` material type has the following characteristics: • Specular Color: shape fill color plus white, which is brightened by 1.5 times the normal value • Specular Power value: 12 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: 4 [*Example*: Consider the following example of the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	`metal` material type: *end example*]
plastic (Plastic)	The `plastic` material type has the following characteristics: • Specular Color: light gray • Specular Power value: 12 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `plastic` material type: *end example*]
powder (Powder)	The `powder` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 10 • Diffuse Color: gray • Ambient Color: gray • Emissive Color: black • Diffuse Fresnel value: 2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: Consider the following example of the `powder` material type: *end example*]
softEdge (Soft Edge)	The `softEdge` material type has the following characteristics: • Specular Color: white • Specular Power value: 35 • Ambient Color: shape fill color • Emissive Color: black • Diffuse Fresnel value: 4 • Alpha Fresnel value: -10 [*Example*: Consider the following example of the `softEdge` material type: *end example*]
softmetal (Soft Metal)	The `softMetal` material type has the following characteristics: • Specular Color: shape fill color lightened with white by 50% • Specular Power value: 8

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	• Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `softmetal` material type: *end example*]
translucentPowder (Translucent Powder)	The `translucentPowder` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 10 • Diffuse Color: shape fill color with 30% transparency • Ambient Color: shape fill color with 30% transparency • Emissive Color: black • Diffuse Fresnel value: 2 • Alpha Fresnel value: -1 [*Example*: Consider the following example of the `translucentPowder` material type: *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
warmMatte (Warm Matte)	The `warmMatte` material type has the following characteristics: • Specular Color: dark gray • Specular Power value: 8 • Ambient Color: shape fill color • Emissive Color: black [*Example*: Consider the following example of the `warmMatte` material type: *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetMaterialType) is located in §A.4.1. *end note*]

20.1.10.51 ST_PresetPatternVal (Preset Pattern Value)

This simple type indicates a preset type of pattern fill. The description of each value contains an illustration of the fill type.

[*Note*: These presets correspond to members of the HatchStyle enumeration in the Microsoft .NET Framework. A reference for this type can be found at http://msdn2.microsoft.com/en-us/library/system.drawing.drawing2d.hatchstyle.aspx. *end note*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cross (Cross)	
dashDnDiag (Dashed Downward Diagonal)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
dashHorz (Dashed Horizontal)	
dashUpDiag (Dashed Upward DIagonal)	
dashVert (Dashed Vertical)	
diagBrick (Diagonal Brick)	
diagCross (Diagonal Cross)	
divot (Divot)	
dkDnDiag (Dark Downward Diagonal)	
dkHorz (Dark Horizontal)	
dkUpDiag (Dark Upward Diagonal)	
dkVert (Dark Vertical)	
dnDiag (Downward Diagonal)	
dotDmnd (Dotted Diamond)	
dotGrid (Dotted Grid)	
horz (Horizontal)	
horzBrick (Horizontal Brick)	
lgCheck (Large Checker Board)	
lgConfetti (Large Confetti)	
lgGrid (Large Grid)	
ltDnDiag (Light Downward Diagonal)	
ltHorz (Light Horizontal)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
ltUpDiag (Light Upward Diagonal)	
ltVert (Light Vertical)	
narHorz (Narrow Horizontal)	
narVert (Narrow Vertical)	
openDmnd (Open Diamond)	
pct10 (10%)	
pct20 (20%)	
pct25 (25%)	
pct30 (30%)	
pct40 (40%)	
pct5 (5%)	
pct50 (50%)	
pct60 (60%)	
pct70 (70%)	
pct75 (75%)	
pct80 (80%)	
pct90 (90%)	
plaid (Plaid)	
shingle (Shingle)	
smCheck (Small Checker Board)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
smConfetti (Small Confetti)	
smGrid (Small Grid)	
solidDmnd (Solid Diamond)	
sphere (Sphere)	
trellis (Trellis)	
upDiag (Upward Diagonal)	
vert (Vertical)	
wave (Wave)	
wdDnDiag (Wide Downward Diagonal)	
wdUpDiag (Wide Upward Diagonal)	
weave (Weave)	
zigZag (Zig Zag)	

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetPatternVal) is located in §A.4.1. *end note*]

20.1.10.52 ST_PresetShadowVal (Preset Shadow Type)

This simple type indicates one of 20 preset shadow types. Each enumeration value description illustrates the type of shadow represented by the value. Each description contains the parameters to the outer shadow effect represented by the preset, in addition to those attributes common to all prstShdw effects.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
shdw1 (Top Left Drop Shadow)	 No additional attributes specified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
shdw10 (Top Left Large Drop Shadow)	 align = "br" sx = 125% sy = 125%
shdw11 (Back Left Long Perspective Shadow)	 align = "b" kx = 40.89° sy = 50%
shdw12 (Back Right Long Perspective Shadow)	 align = "b" kx = -40.89° sy = 50%
shdw13 (Top Left Double Drop Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified. Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dist = 2 * shadow 1's distance
shdw14 (Bottom Right Small Drop Shadow)	 No additional attributes specified.
shdw15 (Front Left Long Perspective Shadow)	 align = "b" kx = 40.89° sy = -50%
shdw16 (Front Right LongPerspective Shadow)	 align = "b" kx = -40.89° sy = -50%
shdw17 (3D Outer Box Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dir = shadow 1's direction + 180°
shdw18 (3D Inner Box Shadow)	 Equivalent to two outer shadow effects. Shadow 1: No additional attributes specified. Shadow 2: color = min(1, shadow 1's color (0 <= r, g, b <= 1) + 102/255), per r, g, b component dir = shadow 1's direction + 180°
shdw19 (Back Center Perspective Shadow)	 align = "b" sy = 50°
shdw2 (Top Right Drop Shadow)	 No additional attributes specified.
shdw20 (Front Bottom Shadow)	 align = "b" sy = -100°
shdw3 (Back Left Perspective Shadow)	 align = "b" ky = 40.89° sy = 50%
shdw4 (Back Right Perspective Shadow)	 align = "b" kx = -40.89° sy = 50%
shdw5 (Bottom Left Drop Shadow)	 No additional attributes specified.
shdw6 (Bottom Right Drop Shadow)	 No additional attributes specified.
shdw7 (Front Left Perspective Shadow)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	align = "b" kx = 40.89° sy = -50%
shdw8 (Front Right Perspective Shadow)	 align = "b" kx = -40.89° sy = -50%
shdw9 (Top Left Small Drop Shadow)	 align = "tl" sx = 75% sy = 75%

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PresetShadowVal) is located in §A.4.1. *end note*]

20.1.10.53 ST_RectAlignment (Rectangle Alignments)

This simple type describes how to position two rectangles relative to each other.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Rectangle Alignment Enum (Bottom))	Bottom
bl (Rectangle Alignment Enum (Bottom Left))	Bottom Left
br (Rectangle Alignment Enum (Bottom Right))	Bottom Right
ctr (Rectangle Alignment Enum (Center))	Center
l (Rectangle Alignment Enum (Left))	Left
r (Rectangle Alignment Enum (Right))	Right
t (Rectangle Alignment Enum (Top))	Top
tl (Rectangle Alignment Enum (Top Left))	Top Left
tr (Rectangle Alignment Enum (Top Right))	Top Right

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RectAlignment) is located in §A.4.1. *end note*]

20.1.10.54 ST_SchemeColorVal (Scheme Color)

This simple type represents a scheme color value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accent1 (Accent Color 1)	Extra scheme color 1
accent2 (Accent Color 2)	Extra scheme color 2
accent3 (Accent Color 3)	Extra scheme color 3
accent4 (Accent Color 4)	Extra scheme color 4
accent5 (Accent Color 5)	Extra scheme color 5
accent6 (Accent Color 6)	Extra scheme color 6
bg1 (Background Color 1)	Semantic background color
bg2 (Background Color 2)	Semantic additional background color
dk1 (Dark Color 1)	Main dark color 1
dk2 (Dark Color 2)	Main dark color 2
folHlink (Followed Hyperlink Color)	Followed Hyperlink Color
hlink (Hyperlink Color)	Regular Hyperlink Color
lt1 (Light Color 1)	Main Light Color 1
lt2 (Light Color 2)	Main Light Color 2
phClr (Style Color)	A color used in theme definitions which means to use the color of the style.
tx1 (Text Color 1)	Semantic text color
tx2 (Text Color 2)	Semantic additional text color

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SchemeColorVal) is located in §A.4.1. *end note*]

20.1.10.55 ST_ShapeID (Shape ID)

Specifies the shape ID for legacy shape identification purposes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShapeID) is located in §A.4.1. *end note*]

20.1.10.56 ST_ShapeType (Preset Shape Types)

This simple type specifies the preset shape geometry that is to be used for a shape. An enumeration of this simple type is used so that a custom geometry does not have to be specified but instead can be constructed automatically by the generating application. For each enumeration listed there is also the corresponding DrawingML code that would be used to construct this shape were it a custom geometry. Within the construction

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

code for each of these preset shapes there are predefined guides that the generating application shall maintain for calculation purposes at all times. The necessary guides should have the following values. Formula syntax components are defined in the fmla attribute of the gd element (§20.1.9.11).

3/4 of a Circle ('3cd4') - Constant value of "16200000.0"

The units here are in 60,000ths of a degree. This is equivalent to 270 degrees.

3/8 of a Circle ('3cd8') - Constant value of "8100000.0"

The units here are in 60,000ths of a degree. This is equivalent to 135 degrees.

5/8 of a Circle ('5cd8') - Constant value of "13500000.0"

The units here are in 60,000ths of a degree. This is equivalent to 225 degrees.

7/8 of a Circle ('7cd8') - Constant value of "18900000.0"

The units here are in 60,000ths of a degree. This is equivalent to 315 degrees.

Shape Bottom Edge ('b') - Constant value of "h"

This is the bottom edge of the shape and since the top edge of the shape is considered the 0 point, the bottom edge is thus the shape height.

1/2 of a Circle ('cd2') - Constant value of "10800000.0"

The units here are in 60,000ths of a degree. This is equivalent to 180 degrees.

1/4 of a Circle ('cd4') - Constant value of "5400000.0"

The units here are in 60,000ths of a degree. This is equivalent to 90 degrees.

1/8 of a Circle ('cd8') - Constant value of "2700000.0"

The units here are in 60,000ths of a degree. This is equivalent to 45 degrees.

Shape Height ('h')

This is the variable height of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

Horizontal Center ('hc') - Calculated value of "*/ w 1.0 2.0"

This is the horizontal center of the shape which is just the width divided by 2.

1/2 of Shape Height ('hd2') - Calculated value of "*/ h 1.0 2.0"

This is 1/2 the shape height.

1/3 of Shape Height ('hd3') - Calculated value of "*/ h 1.0 3.0"

This is 1/3 the shape height.

1/4 of Shape Height ('hd4') - Calculated value of "*/ h 1.0 4.0"

This is 1/4 the shape height.

1/5 of Shape Height ('hd5') - Calculated value of "*/ h 1.0 5.0"

This is 1/5 the shape height.

1/6 of Shape Height ('hd6') - Calculated value of "*/ h 1.0 6.0"

This is 1/6 the shape height.

1/8 of Shape Height ('hd8') - Calculated value of "*/ h 1.0 8.0"

This is 1/8 the shape height.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Shape Left Edge ('l') - Constant value of "0"

This is the left edge of the shape and the left edge of the shape is considered the horizontal 0 point.

Longest Side of Shape ('ls') - Calculated value of "max w h"

This is the longest side of the shape. This value is either the width or the height depending on which is greater.

Shape Right Edge ('r') - Constant value of "w"

This is the right edge of the shape and since the left edge of the shape is considered the 0 point, the right edge is thus the shape width.

Shortest Side of Shape ('ss') - Calculated value of "min w h"

This is the shortest side of the shape. This value is either the width or the height depending on which is smaller.

1/2 Shortest Side of Shape ('ssd2') - Calculated value of "*/ ss 1.0 2.0"

This is 1/2 the shortest side of the shape.

1/4 Shortest Side of Shape ('ssd4') - Calculated value of "*/ ss 1.0 4.0"

This is 1/4 the shortest side of the shape.

1/6 Shortest Side of Shape ('ssd6') - Calculated value of "*/ ss 1.0 6.0"

This is 1/6 the shortest side of the shape.

1/8 Shortest Side of Shape ('ssd8') - Calculated value of "*/ ss 1.0 8.0"

This is 1/8 the shortest side of the shape.

1/16 Shortest Side of Shape ('ssd16') - Calculated value of "*/ ss 1.0 16.0"

This is 1/16 the shortest side of the shape.

1/32 Shortest Side of Shape ('ssd32') - Calculated value of "*/ ss 1.0 32.0"

This is 1/32 the shortest side of the shape.

Shape Top Edge ('t') - Constant value of "0"

This is the top edge of the shape and the top edge of the shape is considered the vertical 0 point.

Vertical Center of Shape ('vc') - Calculated value of "*/ h 1.0 2.0"

This is the vertical center of the shape which is just the height divided by 2.

Shape Width ('w')

This is the variable width of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

1/2 of Shape Width ('wd2') - Calculated value of "*/ w 1.0 2.0"

This is 1/2 the shape width.

1/3 of Shape Width ('wd3') - Calculated value of "*/ w 1.0 3.0"

This is 1/3 the shape width.

1/4 of Shape Width ('wd4') - Calculated value of "*/ w 1.0 4.0"

This is 1/4 the shape width.

1/5 of Shape Width ('wd5') - Calculated value of "*/ w 1.0 5.0"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This is 1/5 the shape width.

1/6 of Shape Width ('wd6') - Calculated value of "*/ w 1.0 6.0"

This is 1/6 the shape width.

1/8 of Shape Width ('wd8') - Calculated value of "*/ w 1.0 8.0"

This is 1/8 the shape width.

1/10 of Shape Width ('wd10') - Calculated value of "*/ w 1.0 10.0"

This is 1/10 the shape width.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
accentBorderCallout1 (Callout 1 with Border and Accent Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
accentBorderCallout2 (Callout 2 with Border and Accent Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
accentBorderCallout3 (Callout 3 with Border and	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
Accent Shape)	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentBorderCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
accentCallout1 (Callout 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
accentCallout2 (Callout 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
accentCallout3 (Callout 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the accentCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonBackPrevious (Back or Previous Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonBackPrevious element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonBeginning (Beginning Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonBeginning element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
actionButtonBlank (Blank Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonBlank element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonDocument (Document Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonDocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonEnd (End Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the actionButtonEnd element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonForwardNext (Forward or Next Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonForwardNext element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonHelp (Help Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonHelp element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
actionButtonHome (Home Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonHome element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonInformation (Information Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonInformation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonMovie (Movie Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonMovie element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
actionButtonReturn (Return Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonReturn element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
actionButtonSound (Sound Button Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the actionButtonSound element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
arc (Curved Arc Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the arc element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentArrow (Bent Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentConnector2 (Bent Connector 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentConnector3 (Bent Connector 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the bentConnector3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentConnector4 (Bent Connector 4 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentConnector5 (Bent Connector 5 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentConnector5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bentUpArrow (Bent Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bentUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bevel (Bevel Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bevel element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
blockArc (Block Arc Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the blockArc element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
borderCallout1 (Callout 1 with Border Shape)	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
borderCallout2 (Callout 2 with Border Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
borderCallout3 (Callout 3 with Border Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the borderCallout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

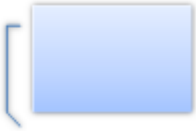
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
bracePair (Brace Pair Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bracePair element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
bracketPair (Bracket Pair Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the bracketPair element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
callout1 (Callout 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
callout2 (Callout 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
callout3 (Callout 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the callout3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
can (Can Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the can element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
chartPlus (Chart Plus Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartPlus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
chartStar (Chart Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartStar element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
chartX (Chart X Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chartX element in the preset shape geometries

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
chevron (Chevron Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chevron element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
chord (Chord Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the chord element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
circularArrow (Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the circularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
cloud (Cloud Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cloud element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
cloudCallout (Callout Cloud Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cloudCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
corner (Corner Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the corner element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
cornerTabs (Corner Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cornerTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
cube (Cube Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the cube element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedConnector2 (Curved Connector 2 Shape)	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedConnector3 (Curved Connector 3 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector3 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedConnector4 (Curved Connector 4 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
curvedConnector5 (Curved Connector 5 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedConnector5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedDownArrow (Curved Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedDownArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedLeftArrow (Curved Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedLeftArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
curvedRightArrow (Curved Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
curvedUpArrow (Curved Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the curvedUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
decagon (Decagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the decagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
diagStripe (Diagonal Stripe Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the diagStripe element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
diamond (Diamond Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the diamond element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

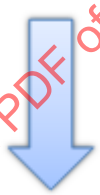
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
dodecagon (Dodecagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the dodecagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
donut (Donut Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the donut element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
doubleWave (Double Wave Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the doubleWave element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
downArrow (Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the downArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
downArrowCallout (Callout Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the downArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
ellipse (Ellipse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipse element in the preset shape geometries

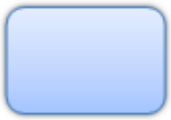
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
ellipseRibbon (Ellipse Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipseRibbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
ellipseRibbon2 (Ellipse Ribbon 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ellipseRibbon2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartAlternateProcess (Alternate Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartAlternateProcess element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartCollate (Collate Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartCollate element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartConnector (Connector Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartConnector element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartDecision (Decision Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDecision element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartDelay (Delay Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDelay element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartDisplay (Display Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartDisplay element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartDocument (Document Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartDocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartExtract (Extract Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartExtract element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartInputOutput (Input Output Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartInputOutput element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartInternalStorage (Internal Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartInternalStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartMagneticDisk (Magnetic Disk Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMagneticDisk element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartMagneticDrum (Magnetic Drum Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMagneticDrum element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartMagneticTape (Magnetic Tape Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartMagneticTape element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartManualInput (Manual Input Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartManualInput element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartManualOperation (Manual Operation Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartManualOperation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartMerge (Merge Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the flowChartMerge element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartMultidocument (Multi-Document Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartMultidocument element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartOfflineStorage (Offline Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOfflineStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartOffpageConnector (Off-Page Connector Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

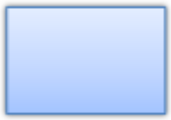
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartOffpageConnector element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartOnlineStorage (Online Storage Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOnlineStorage element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartOr (Or Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartOr element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartPredefinedProcess (Predefined Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartPredefinedProcess element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartPreparation (Preparation Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPreparation element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartProcess (Process Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartProcess element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
flowChartPunchedCard (Punched Card Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

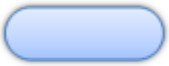
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the flowChartPunchedCard element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartPunchedTape (Punched Tape Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartPunchedTape element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartSort (Sort Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartSort element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartSummingJunction (Summing Junction Flow	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
Shape)	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartSummingJunction element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
flowChartTerminator (Terminator Flow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the flowChartTerminator element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
foldedCorner (Folded Corner Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the foldedCorner element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
frame (Frame Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the frame element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
funnel (Funnel Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the funnel element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
gear6 (Gear 6 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the gear6 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
gear9 (Gear 9 Shape)	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the gear9 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
halfFrame (Half Frame Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the halfFrame element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
heart (Heart Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the heart element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

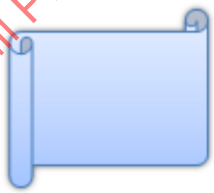
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
heptagon (Heptagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the heptagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
hexagon (Hexagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the hexagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
homePlate (Home Plate Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the homePlate element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
horizontalScroll (Horizontal Scroll Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the horizontalScroll element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
irregularSeal1 (Irregular Seal 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the irregularSeal1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
irregularSeal2 (Irregular Seal 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the irregularSeal2 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftArrow (Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftArrowCallout (Callout Left Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftBrace (Left Brace Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

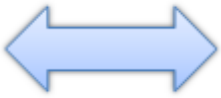
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftBrace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftBracket (Left Bracket Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftBracket element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftCircularArrow (Left Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftCircularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftRightArrow (Left Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall

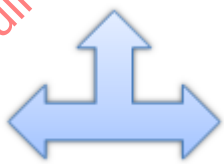
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftRightArrowCallout (Callout Left Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftRightCircularArrow (Left Right Circular Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightCircularArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftRightRibbon (Left Right Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	generate this preset shape definition is contained in the leftRightRibbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
leftRightUpArrow (Left Right Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftRightUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
leftUpArrow (Left Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the leftUpArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
lightningBolt (Lightning Bolt Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

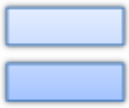
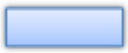
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the lightningBolt element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
line (Line Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the line element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
lineInv (Line Inverse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the lineInv element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
mathDivide (Divide Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathDivide element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
mathEqual (Equal Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathEqual element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathMinus (Minus Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathMinus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
mathMultiply (Multiply Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathMultiply element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
mathNotEqual (Not Equal Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathNotEqual element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
mathPlus (Plus Math Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the mathPlus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
moon (Moon Shape)	Specifies a preset shape geometry. This geometry shall

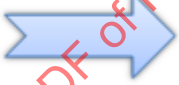
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the moon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
nonIsoscelesTrapezoid (Non-Isosceles Trapezoid Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the nonIsoscelesTrapezoid element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
noSmoking (No Smoking Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the noSmoking element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
notchedRightArrow (Notched Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the notchedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
octagon (Octagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the octagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
parallelogram (Parallelogram Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the parallelogram element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are

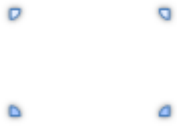
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	described in further detail above. *end note*]
pentagon (Pentagon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pentagon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
pie (Pie Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pie element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
pieWedge (Pie Wedge Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the pieWedge element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
plaque (Plaque Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plaque element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
plaqueTabs (Plaque Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plaqueTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
plus (Plus Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the plus element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
quadArrow (Quad-Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the quadArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
quadArrowCallout (Callout Quad-Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the quadArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
rect (Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
ribbon (Ribbon Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ribbon element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
ribbon2 (Ribbon 2 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the ribbon2 element in the preset shape geometries

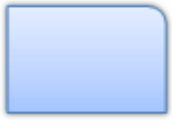
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
rightArrow (Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
rightArrowCallout (Callout Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
rightBrace (Right Brace Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

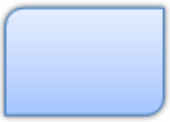
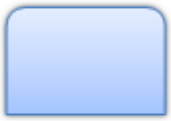
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the rightBrace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
rightBracket (Right Bracket Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rightBracket element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
round1Rect (One Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round1Rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
round2DiagRect (Two Diagonal Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round2DiagRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
round2SameRect (Two Same-side Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the round2SameRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
roundRect (Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the roundRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
rtTriangle (Right Triangle Shape)	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the rtTriangle element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
smileyFace (Smiley Face Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the smileyFace element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
snip1Rect (One Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip1Rect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
snip2DiagRect (Two Diagonal Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip2DiagRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
snip2SameRect (Two Same-side Snip Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snip2SameRect element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
snipRoundRect (One Snip One Round Corner Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the snipRoundRect element in the preset shape geometries electronic addenda of Annex D. The

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	constants used in that markup are guides that are described in further detail above. *end note*]
squareTabs (Square Tabs Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the squareTabs element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star10 (Ten Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star10 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star12 (Twelve Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

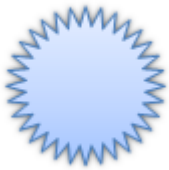
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the star12 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star16 (Sixteen Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star16 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star24 (Twenty Four Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star24 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star32 (Thirty Two Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star32 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
star4 (Four Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star4 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star5 (Five Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star5 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
star6 (Six Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star6 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star7 (Seven Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star7 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
star8 (Eight Pointed Star Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the star8 element in the preset shape geometries electronic addenda of Annex D. The constants used in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	that markup are guides that are described in further detail above. *end note*]
straightConnector1 (Straight Connector 1 Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the straightConnector1 element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
stripedRightArrow (Striped Right Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the stripedRightArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
sun (Sun Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the sun element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
swooshArrow (Swoosh Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the swooshArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
teardrop (Teardrop Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the teardrop element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
trapezoid (Trapezoid Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the trapezoid element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
triangle (Triangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the triangle element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
upArrow (Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

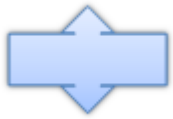
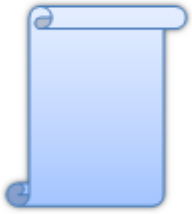
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
upArrowCallout (Callout Up Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
upDownArrow (Up Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the upDownArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
upDownArrowCallout (Callout Up Down Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	the upDownArrowCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
uturnArrow (U-Turn Arrow Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the uturnArrow element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
verticalScroll (Vertical Scroll Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the verticalScroll element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*] 
wave (Wave Shape)	Specifies a preset shape geometry. This geometry shall

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wave element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
wedgeEllipseCallout (Callout Wedge Ellipse Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wedgeEllipseCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
wedgeRectCallout (Callout Wedge Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wedgeRectCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
wedgeRoundRectCallout (Callout Wedge Round Rectangle Shape)	Specifies a preset shape geometry. This geometry shall be designed to match the normative image below. [*Note*: An example of DrawingML which can be used to generate this preset shape definition is contained in the wedgeRoundRectCallout element in the preset shape geometries electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ShapeType) is located in §A.4.1. *end note*]

20.1.10.57 ST_StyleMatrixColumnIndex (Style Matrix Column Index)

This simple type specifies an index into one of the lists in the style matrix specified by the fmtScheme element (bgFillStyleLst, effectStyleLst, fillStyleLst, or lnStyleLst).

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StyleMatrixColumnIndex) is located in §A.4.1. *end note*]

20.1.10.58 ST_SystemColorVal (System Color Value)

This simple type specifies a system color value. This color is based upon the value that this color currently has within the system on which the document is being viewed.

Applications shall use the lastClr attribute to determine the absolute value of the last color used if system colors are not supported.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
3dDkShadow (3D Dark System Color)	Specifies a Dark shadow color for three-dimensional display elements.
3dLight (3D Light System Color)	Specifies a Light color for three-dimensional display elements (for edges facing the light source).
activeBorder (Active Border System Color)	Specifies an Active Window Border Color.
activeCaption (Active Caption System Color)	Specifies the active window title bar color. In particular the left side color in the color gradient of an active window's title bar if the gradient effect is enabled.
appWorkspace (Application Workspace System Color)	Specifies the Background color of multiple document interface (MDI) applications.
background (Background System Color)	Specifies the desktop background color.
btnFace (Button Face System Color)	Specifies the face color for three-dimensional display elements and for dialog box backgrounds.
btnHighlight (Button Highlight System Color)	Specifies the highlight color for three-dimensional display elements (for edges facing the light source).
btnShadow (Button Shadow System Color)	Specifies the shadow color for three-dimensional display elements (for edges facing away from the light source).
btnText (Button Text System Color)	Specifies the color of text on push buttons.
captionText (Caption Text System Color)	Specifies the color of text in the caption, size box, and scroll bar arrow box.
gradientActiveCaption (Gradient Active Caption System Color)	Specifies the right side color in the color gradient of an active window's title bar.
gradientInactiveCaption (Gradient Inactive Caption System Color)	Specifies the right side color in the color gradient of an inactive window's title bar.
grayText (Gray Text System Color)	Specifies a grayed (disabled) text. This color is set to 0 if the current display driver does not support a solid gray color.
highlight (Highlight System Color)	Specifies the color of Item(s) selected in a control.
highlightText (Highlight Text System Color)	Specifies the text color of item(s) selected in a control.
hotLight (Hot Light System Color)	Specifies the color for a hyperlink or hot-tracked item.
inactiveBorder (Inactive Border System Color)	Specifies the color of the Inactive window border.
inactiveCaption (Inactive Caption System Color)	Specifies the color of the Inactive window caption. Specifies the left side color in the color gradient of an inactive window's title bar if the gradient effect is enabled.
inactiveCaptionText (Inactive Caption Text System Color)	Specifies the color of text in an inactive caption.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
infoBk (Info Back System Color)	Specifies the background color for tooltip controls.
infoText (Info Text System Color)	Specifies the text color for tooltip controls.
menu (Menu System Color)	Specifies the menu background color.
menuBar (Menu Bar System Color)	Specifies the background color for the menu bar when menus appear as flat menus.
menuHighlight (Menu Highlight System Color)	Specifies the color used to highlight menu items when the menu appears as a flat menu.
menuText (Menu Text System Color)	Specifies the color of Text in menus.
scrollBar (Scroll Bar System Color)	Specifies the scroll bar gray area color.
window (Window System Color)	Specifies window background color.
windowFrame (Window Frame System Color)	Specifies the window frame color.
windowText (Window Text System Color)	Specifies the color of text in windows.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SystemColorVal) is located in §A.4.1. *end note*]

20.1.10.59 ST_TextAlignType (Text Alignment Types)

This simple type specifies the text alignment types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Text Alignment Enum (Center))	Align text in the center.
dist (Text Alignment Enum (Distributed))	Distributes the text words across an entire text line.
just (Text Alignment Enum (Justified))	Align text so that it is justified across the whole line. It is smart in the sense that it does not justify sentences which are short.
justLow (Text Alignment Enum (Justified Low))	Aligns the text with an adjusted kashida length for Arabic text.
l (Text Alignment Enum (Left))	Align text to the left margin.
r (Text Alignment Enum (Right))	Align text to the right margin.
thaiDist (Text Alignment Enum (Thai Distributed))	Distributes Thai text specially, because each character is treated as a word.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAlignType) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.1.10.60 ST_TextAnchoringType (Text Anchoring Types)

This simple type specifies a list of available anchoring types for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Text Anchor Enum (Bottom))	Anchor the text at the bottom of the bounding rectangle.
ctr (Text Anchor Enum (Center))	Anchor the text at the middle of the bounding rectangle.
dist (Text Anchor Enum (Distributed))	Anchor the text so that it is distributed vertically. When text is horizontal, this spaces out the actual lines of text and is almost always identical in behavior to anchorJustified (special case: if only 1 line, then anchored in middle). When text is vertical, then it distributes the letters vertically. This is different than anchorJustified, because it always forces distribution of the words, even if there are only one or two words in a line.
just (Text Anchor Enum (Justified))	Anchor the text so that it is justified vertically. When text is horizontal, this spaces out the actual lines of text and is almost always identical in behavior to 'distrib' (special case: if only 1 line, then anchored at top). When text is vertical, then it justifies the letters vertically. This is different than anchorDistributed, because in some cases such as very little text in a line, it does not justify.
t (Text Anchoring Type Enum (Top))	Anchor the text at the top of the bounding rectangle.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchoringType) is located in §A.4.1. *end note*]

20.1.10.61 ST_TextAutonumberScheme (Text Auto-number Schemes)

This simple type specifies a list of automatic numbering schemes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alphaLcParenBoth (Autonumber Enum (alphaLcParenBoth))	(a), (b), (c), …

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
alphaLcParenR (Autonumbering Enum (alphaLcParenR))	a), b), c), …
alphaLcPeriod (Autonumbering Enum (alphaLcPeriod))	a., b., c., …
alphaUcParenBoth (Autonumbering Enum (alphaUcParenBoth))	(A), (B), (C), …
alphaUcParenR (Autonumbering Enum (alphaUcParenR))	A), B), C), …
alphaUcPeriod (Autonumbering Enum (alphaUcPeriod))	A., B., C., …
arabic1Minus (Autonumbering Enum (arabic1Minus))	Bidi Arabic 1 (AraAlpha) with ANSI minus symbol
arabic2Minus (Autonumbering Enum (arabic2Minus))	Bidi Arabic 2 (AraAbjad) with ANSI minus symbol
arabicDbPeriod (Autonumbering Enum (arabicDbPeriod))	Dbl-byte Arabic numbers w/ double-byte period
arabicDbPlain (Autonumbering Enum (arabicDbPlain))	Dbl-byte Arabic numbers
arabicParenBoth (Autonumbering Enum (arabicParenBoth))	(1), (2), (3), …
arabicParenR (Autonumbering Enum (arabicParenR))	1), 2), 3), …
arabicPeriod (Autonumbering Enum (arabicPeriod))	1., 2., 3., …
arabicPlain (Autonumbering Enum (arabicPlain))	1, 2, 3, …
circleNumDbPlain (Autonumbering Enum (circleNumDbPlain))	Dbl-byte circle numbers (1-10 circle[0x2460-], 11-arabic numbers)
circleNumWdBlackPlain (Autonumbering Enum (circleNumWdBlackPlain))	Wingdings black circle numbers
circleNumWdWhitePlain (Autonumbering Enum (circleNumWdWhitePlain))	Wingdings white circle numbers (0-10 circle[0x0080-], 11- arabic numbers)
ea1ChsPeriod (Autonumbering Enum (ea1ChsPeriod))	EA: Simplified Chinese w/ single-byte period
ea1ChsPlain (Autonumbering Enum (ea1ChsPlain))	EA: Simplified Chinese (TypeA 1-99, TypeC 100-)
ea1ChtPeriod (Autonumbering Enum (ea1ChtPeriod))	EA: Traditional Chinese w/ single-byte period
ea1ChtPlain (Autonumbering Enum (ea1ChtPlain))	EA: Traditional Chinese (TypeA 1-19, TypeC 20-)
ea1JpnChsDbPeriod (Autonumbering Enum (ea1JpnChsDbPeriod))	EA: Japanese w/ double-byte period
ea1JpnKorPeriod (Autonumbering Enum (	EA: Japanese/Korean w/ single-byte period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
ea1JpnKorPeriod))	
ea1JpnKorPlain (Autonumbering Enum (ea1JpnKorPlain))	EA: Japanese/Korean (TypeC 1-)
hebrew2Minus (Autonumbering Enum (hebrew2Minus))	Bidi Hebrew 2 with ANSI minus symbol
hindiAlpha1Period (Autonumbering Enum (hindiAlpha1Period))	Hindi alphabet period - consonants
hindiAlphaPeriod (Autonumbering Enum (hindiAlphaPeriod))	Hindi alphabet period - vowels
hindiNumParenR (Autonumbering Enum (hindiNumParenR))	Hindi numerical parentheses - right
hindiNumPeriod (Autonumbering Enum (hindiNumPeriod))	Hindi numerical period
romanLcParenBoth (Autonumbering Enum (romanLcParenBoth))	(i), (ii), (iii), …
romanLcParenR (Autonumbering Enum (romanLcParenR))	i), ii), iii), …
romanLcPeriod (Autonumbering Enum (romanLcPeriod))	i., ii., iii., …
romanUcParenBoth (Autonumbering Enum (romanUcParenBoth))	(I), (II), (III), …
romanUcParenR (Autonumbering Enum (romanUcParenR))	I), II), III), …
romanUcPeriod (Autonumbering Enum (romanUcPeriod))	I., II., III., …
thaiAlphaParenBoth (Autonumbering Enum (thaiAlphaParenBoth))	Thai alphabet parentheses - both
thaiAlphaParenR (Autonumbering Enum (thaiAlphaParenR))	Thai alphabet parentheses - right
thaiAlphaPeriod (Autonumbering Enum (thaiAlphaPeriod))	Thai alphabet period
thaiNumParenBoth (Autonumbering Enum (thaiNumParenBoth))	Thai numerical parentheses - both
thaiNumParenR (Autonumbering Enum (thaiNumParenR))	Thai numerical parentheses - right
thaiNumPeriod (Autonumbering Enum (thaiNumPeriod))	Thai numerical period

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAutonumberScheme) is located in §A.4.1. *end note*]

20.1.10.62 ST_TextBulletSizePercent (Bullet Size Percentage)

This simple type specifies the range that the bullet percent can be. A bullet percent is the size of the bullet with respect to the text that should follow it.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: 0*((2[5-9])|([3-9][0-9])|([1-3][0-9][0-9])|400)%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletSizePercent) is located in §A.4.1. *end note*]

20.1.10.63 ST_TextBulletStartAtNum (Start Bullet At Number)

This simple type specifies the range that the start at number for a bullet's auto-numbering sequence can begin at. When the numbering is alphabetical, then the numbers map to the appropriate letter. 1->a, 2->b, etc. If the numbers go above 26, then the numbers begin to double up. For example, 27->aa and 53->aaa.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 32767.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletStartAtNum) is located in §A.4.1. *end note*]

20.1.10.64 ST_TextCapsType (Text Cap Types)

This simple type specifies the cap types of the text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (Text Caps Enum (All))	Apply all caps on the text. All lower case letters are converted to upper case even though they are stored differently in the backing store.
none (Text Caps Enum (None))	The reason we cannot implicitly have noCaps be the scenario where capitalization is not specified is because not being specified implies deriving from a particular style and the user might want to override that and make some text not have a capitalization

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	scheme even though the style says otherwise.
small (Text Caps Enum (Small))	Apply small caps to the text. All letters are converted to lower case.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextCapsType) is located in §A.4.1. *end note*]

20.1.10.65 ST_TextColumnCount (Text Column Count)

This simple type specifies the number of columns.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 16.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextColumnCount) is located in §A.4.1. *end note*]

20.1.10.66 ST_TextFontAlignType (Font Alignment Types)

This simple type specifies the different kinds of font alignment.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Font Alignment Enum (Automatic))	When the text flow is horizontal or simple vertical same as fontBaseline but for other vertical modes same as fontCenter.
b (Font Alignment Enum (Bottom))	The letters are anchored to the very bottom of a single line. This is different than the bottom baseline because of letters such as "g," "q," "y," etc.
base (Font Alignment Enum (Baseline))	The letters are anchored to the bottom baseline of a single line.
ctr (Font Alignment Enum (Center))	The letters are anchored between the two baselines of a single line.
t (Font Alignment Enum (Top))	The letters are anchored to the top baseline of a single line.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontAlignType) is located in §A.4.1. *end note*]

20.1.10.67 ST_TextFontScalePercentOrPercentString (Text Font Scale Percentage)

This simple type specifies that its contents will contain a text font scale percent percentage. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontScalePercentOrPercentString) is located in §A.4.1. *end note*]

20.1.10.68 ST_TextFontSize (Text Font Size)

This simple type specifies the size of any text in hundredths of a point. Shall be at least 1 point.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 100.
- This simple type has a maximum value of less than or equal to 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextFontSize) is located in §A.4.1. *end note*]

20.1.10.69 ST_TextHorzOverflowType (Text Horizontal Overflow Types)

This simple type specifies the text horizontal overflow types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clip (Text Horizontal Overflow Enum (Clip))	When a big character does not fit into a line, clip it at the proper horizontal overflow.
overflow (Text Horizontal Overflow Enum (Overflow))	When a big character does not fit into a line, allow a horizontal overflow.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextHorzOverflowType) is located in §A.4.1. *end note*]

20.1.10.70 ST_TextIndent (Text Indentation)

This simple type specifies the text indentation amount to be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The units of measurement used here are EMUs (English Metric Units).

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -51206400.
- This simple type has a maximum value of less than or equal to 51206400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextIndent) is located in §A.4.1. *end note*]

20.1.10.71 ST_TextIndentLevelType (Text Indent Level Type)

This simple type specifies the indent level type. We support list level 0 to 8, and we use -1 and -2 for outline mode levels that should only exist in memory.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 8.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextIndentLevelType) is located in §A.4.1. *end note*]

20.1.10.72 ST_TextMargin (Text Margin)

This simple type specifies the margin that is used and its corresponding size.

This simple type's contents are a restriction of the ST_Coordinate32Unqualified datatype (§20.1.10.18).

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 51206400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextMargin) is located in §A.4.1. *end note*]

20.1.10.73 ST_TextNonNegativePoint (Text Non-Negative Point)

This simple type specifies a non-negative font size in hundredths of a point. This is restricted to the range [0, 400000].

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextNonNegativePoint) is located in §A.4.1. *end note*]

20.1.10.74 ST_TextPoint (Text Point)

This simple type specifies a coordinate within the document. This can be used for measurements or spacing; its maximum size is +/- 4000 points.

Its contents can contain either:

- A whole number, whose contents consist of a measurement in hundredths of a point
- A number immediately followed by a unit identifier

This simple type is a union of the following types:

- The ST_TextPointUnqualified simple type (§20.1.10.75).
- The ST_UniversalMeasure simple type (§22.9.2.15).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextPoint) is located in §A.4.1. *end note*]

20.1.10.75 ST_TextPointUnqualified (Text Point)

This simple type specifies a font size in hundredths of a point. This is restricted to the range [-400000, 400000], i.e from -4000 pt to 4000 pt.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -400000.
- This simple type has a maximum value of less than or equal to 400000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextPointUnqualified) is located in §A.4.1. *end note*]

20.1.10.76 ST_TextShapeType (Preset Text Shape Types)

This simple type specifies the preset text shape geometry that is to be used for a shape. An enumeration of this simple type is used so that a custom geometry does not have to be specified but instead can be constructed automatically by the generating application. For each enumeration listed there is also the corresponding DrawingML code that would be used to construct this shape were it a custom geometry. Within the construction code for each of these preset text shapes there are predefined guides that the generating application shall maintain for calculation purposes at all times. The necessary guides should have the following values. Formula syntax components are defined in the fmla attribute of the gd element (§20.1.9.11).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

3/4 of a Circle ('3cd4') - Constant value of "16200000.0"

The units here are in 60,000ths of a degree. This is equivalent to 270 degrees.

3/8 of a Circle ('3cd8') - Constant value of "8100000.0"

The units here are in 60,000ths of a degree. This is equivalent to 135 degrees.

5/8 of a Circle ('5cd8') - Constant value of "13500000.0"

The units here are in 60,000ths of a degree. This is equivalent to 225 degrees.

7/8 of a Circle ('7cd8') - Constant value of "18900000.0"

The units here are in 60,000ths of a degree. This is equivalent to 315 degrees.

Shape Bottom Edge ('b') - Constant value of "h"

This is the bottom edge of the shape and since the top edge of the shape is considered the 0 point, the bottom edge is thus the shape height.

1/2 of a Circle ('cd2') - Constant value of "10800000.0"

The units here are in 60,000ths of a degree. This is equivalent to 180 degrees.

1/4 of a Circle ('cd4') - Constant value of "5400000.0"

The units here are in 60,000ths of a degree. This is equivalent to 90 degrees.

1/8 of a Circle ('cd8') - Constant value of "2700000.0"

The units here are in 60,000ths of a degree. This is equivalent to 45 degrees.

Shape Height ('h')

This is the variable height of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

Horizontal Center ('hc') - Calculated value of "*/ w 1.0 2.0"

This is the horizontal center of the shape which is just the width divided by 2.

1/2 of Shape Height ('hd2') - Calculated value of "*/ h 1.0 2.0"

This is 1/2 the shape height.

1/3 of Shape Height ('hd3') - Calculated value of "*/ h 1.0 3.0"

This is 1/3 the shape height.

1/4 of Shape Height ('hd4') - Calculated value of "*/ h 1.0 4.0"

This is 1/4 the shape height.

1/5 of Shape Height ('hd5') - Calculated value of "*/ h 1.0 5.0"

This is 1/5 the shape height.

1/6 of Shape Height ('hd6') - Calculated value of "*/ h 1.0 6.0"

This is 1/6 the shape height.

1/8 of Shape Height ('hd8') - Calculated value of "*/ h 1.0 8.0"

This is 1/8 the shape height.

Shape Left Edge ('l') - Constant value of "0"

This is the left edge of the shape and the left edge of the shape is considered the horizontal 0 point.

Longest Side of Shape ('ls') - Calculated value of "max w h"

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This is the longest side of the shape. This value is either the width or the height depending on which is greater.

Shape Right Edge ('r') - Constant value of "w"

This is the right edge of the shape and since the left edge of the shape is considered the 0 point, the right edge is thus the shape width.

Shortest Side of Shape ('ss') - Calculated value of "min w h"

This is the shortest side of the shape. This value is either the width or the height depending on which is smaller.

1/2 Shortest Side of Shape ('ssd2') - Calculated value of "*/ ss 1.0 2.0"

This is 1/2 the shortest side of the shape.

1/4 Shortest Side of Shape ('ssd4') - Calculated value of "*/ ss 1.0 4.0"

This is 1/4 the shortest side of the shape.

1/6 Shortest Side of Shape ('ssd6') - Calculated value of "*/ ss 1.0 6.0"

This is 1/6 the shortest side of the shape.

1/8 Shortest Side of Shape ('ssd8') - Calculated value of "*/ ss 1.0 8.0"

This is 1/8 the shortest side of the shape.

1/16 Shortest Side of Shape ('ssd16') - Calculated value of "*/ ss 1.0 16.0"

This is 1/16 the shortest side of the shape.

1/32 Shortest Side of Shape ('ssd32') - Calculated value of "*/ ss 1.0 32.0"

This is 1/32 the shortest side of the shape.

Shape Top Edge ('t') - Constant value of "0"

This is the top edge of the shape and the top edge of the shape is considered the vertical 0 point.

Vertical Center of Shape ('vc') - Calculated value of "*/ h 1.0 2.0"

This is the vertical center of the shape which is just the height divided by 2.

Shape Width ('w')

This is the variable width of the shape defined in the shape properties. This value is received from the shape transform listed within the <spPr> element.

1/2 of Shape Width ('wd2') - Calculated value of "*/ w 1.0 2.0"

This is 1/2 the shape width.

1/3 of Shape Width ('wd3') - Calculated value of "*/ w 1.0 3.0"

This is 1/3 the shape width.

1/4 of Shape Width ('wd4') - Calculated value of "*/ w 1.0 4.0"

This is 1/4 the shape width.

1/5 of Shape Width ('wd5') - Calculated value of "*/ w 1.0 5.0"

This is 1/5 the shape width.

1/6 of Shape Width ('wd6') - Calculated value of "*/ w 1.0 6.0"

This is 1/6 the shape width.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

1/8 of Shape Width ('wd8') - Calculated value of "*/ w 1.0 8.0"

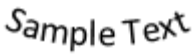
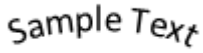
This is 1/8 the shape width.

1/10 of Shape Width ('wd10') - Calculated value of "*/ w 1.0 10.0"

This is 1/10 the shape width.

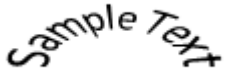
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
textArchDown (Downward Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textArchDownPour (Downward Pour Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchDownPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textArchUp (Upward Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail

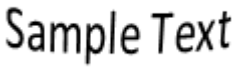
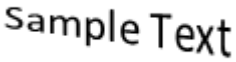
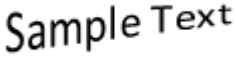
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	above. *end note*]
textArchUpPour (Upward Pour Arch Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textArchUpPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textButton (Button Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textButton element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textButtonPour (Button Pour Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textButtonPour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

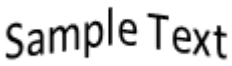
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
textCanDown (Downward Can Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCanDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCanUp (Upward Can Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCanUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCascadeDown (Downward Cascade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCascadeDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCascadeUp (Upward Cascade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can

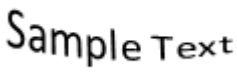
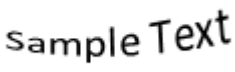
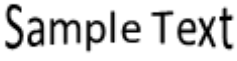
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	be used to achieve this effect is contained in the textCascadeUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textChevron (Chevron Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textChevron element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textChevronInverted (Inverted Chevron Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textChevronInverted element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCircle (Circle Text Shape)	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCircle element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCirclePour (Circle Pour Text Shape)	

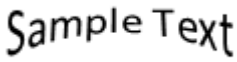
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCirclePour element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCurveDown (Downward Curve Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCurveDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textCurveUp (Upward Curve Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textCurveUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflate (Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail

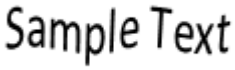
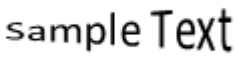
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	above. *end note*]
textDeflateBottom (Bottom Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateBottom element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateInflate (Deflate-Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateInflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateInflateDeflate (Deflate-Inflate-Deflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateInflateDeflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDeflateTop (Top Deflate Text Shape)	Specifies a text shape that shall match the normative shape shown above.

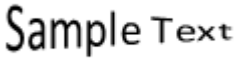
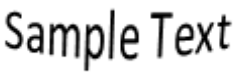
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDeflateTop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textDoubleWave1 (Double Wave 1 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textDoubleWave1 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeDown (Downward Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeLeft (Left Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeLeft element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
textFadeRight (Right Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeRight element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textFadeUp (Upward Fade Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textFadeUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflate (Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textInflate element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflateBottom (Bottom Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the

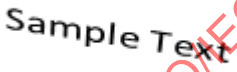
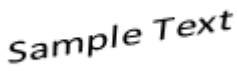
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

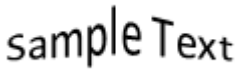
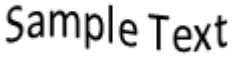
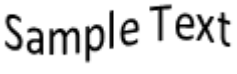
Enumeration Value	Description
	textInflateBottom element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textInflateTop (Top Inflate Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textInflateTop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textNoShape (No Text Shape)	Specifies that the text has no associated shape with it and thus the text should not be warped but instead be constrained by the normal text bounding box.
textPlain (Plain Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textPlain element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textRingInside (Inside Ring Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textRingInside element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

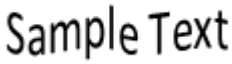
Enumeration Value	Description
textRingOutside (Outside Ring Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textRingOutside element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textSlantDown (Downward Slant Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textSlantDown element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textSlantUp (Upward Slant Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textSlantUp element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textStop (Stop Sign Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	be used to achieve this effect is contained in the textStop element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textTriangle (Triangle Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textTriangle element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textTriangleInverted (Inverted Triangle Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textTriangleInverted element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textWave1 (Wave 1 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave1 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textWave2 (Wave 2 Text Shape)	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave2 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]
textWave4 (Wave 4 Text Shape)	 Specifies a text shape that shall match the normative shape shown above. [*Note*: An example of DrawingML markup which can be used to achieve this effect is contained in the textWave4 element in the preset text warp electronic addenda of Annex D. The constants used in that markup are guides that are described in further detail above. *end note*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextShapeType) is located in §A.4.1. *end note*]

20.1.10.77 ST_TextSpacingPercentOrPercentString (Text Spacing Percent)

This simple type specifies that its contents will contain a text font spacing percentage. See the union's member types for details.

This simple type is a union of the following types:

- The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextSpacingPercentOrPercentString) is located in §A.4.1. *end note*]

20.1.10.78 ST_TextSpacingPoint (Text Spacing Point)

This simple type specifies the Text Spacing that is used in terms of font point size.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- This simple type has a maximum value of less than or equal to 158400.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextSpacingPoint) is located in §A.4.1. *end note*]

20.1.10.79 ST_TextStrikeType (Text Strike Type)

This simple type specifies the strike type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dblStrike (Text Strike Enum (Double Strike))	A double strikethrough applied on the text
noStrike (Text Strike Enum (No Strike))	No strike is applied to the text
sngStrike (Text Strike Enum (Single Strike))	A single strikethrough is applied to the text

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextStrikeType) is located in §A.4.1. *end note*]

20.1.10.80 ST_TextTabAlignType (Text Tab Alignment Types)

This simple type specifies the text tab alignment types.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Text Tab Alignment Enum (Center))	The text at this tab stop is center aligned.
dec (Text Tab Alignment Enum (Decimal))	At this tab stop, the decimals are lined up. From a user's point of view, the text here behaves as right aligned until the decimal, and then as left aligned after the decimal.
l (Text Tab Alignment Enum (Left))	The text at this tab stop is left aligned.
r (Text Tab Alignment Enum (Right))	The text at this tab stop is right aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextTabAlignType) is located in §A.4.1. *end note*]

20.1.10.81 ST_TextTypeface (Text Typeface)

This simple type specifies the way we represent a font typeface.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextTypeface) is located in §A.4.1. *end note*]

20.1.10.82 ST_TextUnderlineType (Text Underline Types)

This simple type specifies the text underline types that is used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
dash (Text Underline Enum (Dashed))	Underline the text with a single, dashed line of normal thickness.
dashHeavy (Text Underline Enum (Heavy Dashed))	Underline the text with a single, dashed, thick line.
dashLong (Text Underline Enum (Long Dashed))	Underline the text with a single line consisting of long dashes of normal thickness.
dashLongHeavy (Text Underline Enum (Heavy Long Dashed))	Underline the text with a single line consisting of long, thick dashes.
dbl (Text Underline Enum (Double))	Underline the text with two lines of normal thickness.
dotDash (Text Underline Enum (Dot Dash))	Underline the text with a single line of normal thickness consisting of repeating dots and dashes.
dotDashHeavy (Text Underline Enum (Heavy Dot Dash))	Underline the text with a single, thick line consisting of repeating dots and dashes.
dotDotDash (Text Underline Enum (Dot Dot Dash))	Underline the text with a single line of normal thickness consisting of repeating two dots and dashes.
dotDotDashHeavy (Text Underline Enum (Heavy Dot Dot Dash))	Underline the text with a single, thick line consisting of repeating two dots and dashes.
dotted (Text Underline Enum (Dotted))	Underline the text with a single, dotted line of normal thickness.
dottedHeavy (Text Underline Enum (Heavy Dotted))	Underline the text with a single, thick, dotted line.
heavy (Text Underline Enum (Heavy))	Underline the text with a single, thick line.
none (Text Underline Enum (None))	The reason we cannot implicitly have noUnderline be the scenario where underline is not specified is because not being specified implies deriving from a particular style and the user might want to override that and make some text not be underlined even though the style says otherwise.
sng (Text Underline Enum (Single))	Underline the text with a single line of normal thickness.
wavy (Text Underline Enum (Wavy))	Underline the text with a single wavy line of normal thickness.
wavyDbl (Text Underline Enum (Double Wavy))	Underline the text with two wavy lines of normal

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	thickness.
wavyHeavy (Text Underline Enum (Heavy Wavy))	Underline the text with a single, thick wavy line.
words (Text Underline Enum (Words))	Underline just the words and not the spaces between them.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextUnderlineType) is located in §A.4.1. *end note*]

20.1.10.83 ST_TextVerticalType (Vertical Text Types)

If there is vertical text, determines what kind of vertical text is going to be used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
eaVert (Vertical Text Type Enum (East Asian Vertical))	A special version of vertical text, where some fonts are displayed as if rotated by 90 degrees while some fonts (mostly East Asian) are displayed vertical.
horz (Vertical Text Type Enum (Horizontal))	Horizontal text. This should be default.
mongolianVert (Vertical Text Type Enum (Mongolian Vertical))	A special version of vertical text, where some fonts are displayed as if rotated by 90 degrees while some fonts (mostly East Asian) are displayed vertical. The difference between this and the eastAsianVertical is the text flows top down then LEFT RIGHT, instead of RIGHT LEFT
vert (Vertical Text Type Enum (Vertical))	Determines if all of the text is vertical orientation (each line is 90 degrees rotated clockwise, so it goes from top to bottom; each next line is to the left from the previous one).
vert270 (Vertical Text Type Enum (Vertical 270))	Determines if all of the text is vertical orientation (each line is 270 degrees rotated clockwise, so it goes from bottom to top; each next line is to the right from the previous one).
wordArtVert (Vertical Text Type Enum (WordArt Vertical))	Determines if all of the text is vertical ("one letter on top of another").
wordArtVertRtl (Vertical WordArt Right to Left)	Specifies that vertical WordArt should be shown from right to left rather than left to right.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextVerticalType) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.1.10.84 ST_TextVertOverflowType (Text Vertical Overflow)

This simple type specifies the text vertical overflow.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clip (Text Overflow Enum (Clip))	Pay attention to top and bottom barriers. Provide no indication that there is text which is not visible.
ellipsis (Text Overflow Enum (Ellipsis))	Pay attention to top and bottom barriers. Use an ellipsis to denote that there is text which is not visible.
overflow (Text Overflow Enum (Overflow))	Overflow the text and pay no attention to top and bottom barriers.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextVertOverflowType) is located in §A.4.1. *end note*]

20.1.10.85 ST_TextWrappingType (Text Wrapping Types)

Text Wrapping Types

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (Text Wrapping Type Enum (None))	No wrapping occurs on this text body. Words spill out without paying attention to the bounding rectangle boundaries.
square (Text Wrapping Type Enum (Square))	Determines whether we wrap words within the bounding rectangle.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextWrappingType) is located in §A.4.1. *end note*]

20.1.10.86 ST_TileFlipMode (Tile Flip Mode)

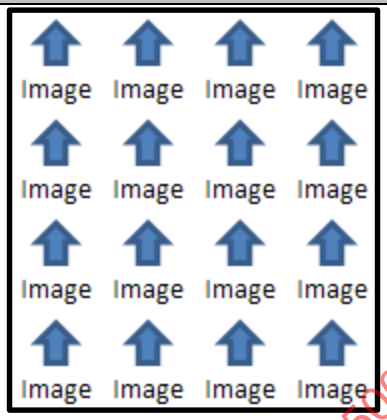
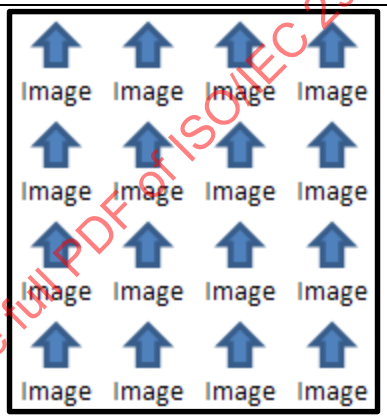
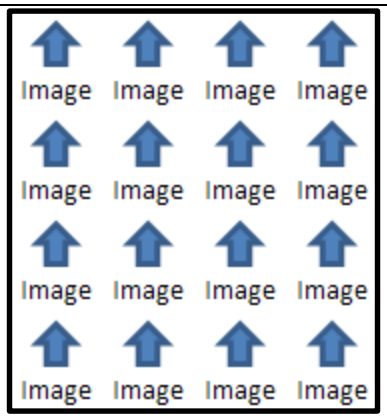
This simple type indicates whether/how to flip the contents of a tile region when using it to fill a larger fill region.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

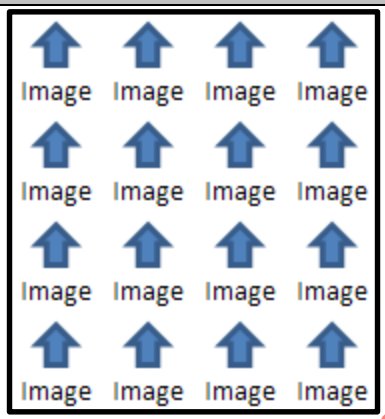
©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
none (None)	 Tiles are not flipped.
x (Horizontal)	 Tiles are flipped horizontally.
xy (Horizontal and Vertical)	 Tiles are flipped both horizontally and vertically.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
y (Vertical)	 Tiles are flipped vertically.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TileFlipMode) is located in §A.4.1. *end note*]

20.1.10.87 ST_TextBulletSize (Bullet Size Percentage)

This simple type specifies the range that the bullet percent can be. A bullet percent is the size of the bullet with respect to the text that should follow it, with a minimum size of 25% and maximum size of 400%.

This simple type is a union of the following types:

- The ST_TextBulletSizePercent simple type (§20.1.10.62)

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBulletSize) is located in § A.4.1. *end note*]

20.2 DrawingML - Picture

These elements encompass the definition of pictures within the DrawingML framework. While pictures are in many ways very similar to shapes they have specific properties that are unique in order to optimize for picture-specific scenarios. Some of these properties include Fill behavior, Border behavior and Resize behavior.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.2.1 Table of Contents

This subclause is informative.

20.2.2 Elements 3088
20.2.2.1 blipFill (Picture Fill) 3088
20.2.2.2 cNvPicPr (Non-Visual Picture Drawing Properties) 3091
20.2.2.3 cNvPr (Non-Visual Drawing Properties) 3091
20.2.2.4 nvPicPr (Non-Visual Picture Properties) 3094
20.2.2.5 pic (Picture) 3094
20.2.2.6 spPr (Shape Properties) 3095

End of informative text.

20.2.2 Elements

The following section defines the Picture portion of the DrawingML framework.

20.2.2.1 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  …
</pic:pic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  <pic:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
          </a:gs>
        </a:gsLst>
      </a:gradFill>
    </pic:spPr>
    …
  </pic:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.2.2.2 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="Lilly.jpg"/>
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </p:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

20.2.2.3 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    <p:cNvPr id="4" name="Lilly.jpg"/>
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text which can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique	Specifies a unique identifier for the current DrawingML object within the current

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of 10, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of foo.jpg, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.2.2.4 nvPicPr (Non-Visual Picture Properties)

This element specifies the non visual properties for a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    …
  </pic:nvPicPr>
  …
</pic:pic>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.4.2. *end note*]

20.2.2.5 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following DrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<pic:pic>
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </pic:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  <pic:blipFill>
  …
  </pic:blipFill>
  <pic:spPr>
  …
  </pic:spPr>
</pic:pic>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.2.2.6 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe the visual appearance of a picture within a document. This allows for a picture to have both the properties of a shape as well as picture specific properties that are allowed under the pic element.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.3 DrawingML - Locked Canvas

Within a DrawingML object, a *locked canvas* allows DrawingML objects to be placed in a format where they can be viewed but not edited by the hosting application. This allows DrawingML objects not supported by an application to be included and viewed in applications where they cannot be edited.

20.3.1 Table of Contents

This subclause is informative.

20.3.2 Basics 3095
20.3.2.1 lockedCanvas (Locked Canvas Container) 3095

End of informative text.

20.3.2 Basics

This section specifies a locked canvas within the basic DrawingML framework.

20.3.2.1 lockedCanvas (Locked Canvas Container)

The locked canvas element acts as a container for more advanced drawing objects. The notion of a locked canvas comes from the fact that the generating application opening the file cannot create this object and can

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

thus not perform edits either. Thus the drawing object is locked from all UI adjustments that would normally take place.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GvmlGroupShape) is located in §A.4.1. *end note*]

20.4 DrawingML - WordprocessingML Drawing

Within a WordprocessingML document, it is possible to include graphical DrawingML objects:

- Pictures (§20.2)
- Locked Canvases (§20.3)
- Diagrams (§21.4)
- Charts (§21.2)

When these objects are present in a word processing document, it is necessary to include information which specifies how the objects shall be positioned relative to the paginated document. [*Example*: Whether the object is displayed in line with text. *end example*]

The WordprocessingML Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a word processing document.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<w:r>
  <w:drawing>
    <wp:anchor relativeHeight="10" allowOverlap="true">
      <wp:positionH relativeFrom="margin">
        <wp:align>center</wp:align>
      </wp:positionH>
      <wp:positionV relativeFrom="margin">
        <wp:align>center</wp:align>
      </wp:positionV>
      <wp:extent cx="2441542" cy="1828800"/>
      <wp:wrapSquare wrapText="bothSides"/>
      <a:graphic>
        …
      </a:graphic>
    </wp:anchor>
  </w:drawing>
</w:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The anchor element (§20.4.2.3) specifies that this object is not positioned in line with text, and its child elements specify that the object is centered on the page horizontally and vertically (§20.4.2.10; §20.4.2.11), and that text can wrap around it in a square (§20.4.2.17). *end example*]

20.4.1 Table of Contents

This subclause is informative.

20.4.2 Elements 3098

20.4.2.1 align (Relative Horizontal Alignment) 3098
20.4.2.2 align (Relative Vertical Alignment) 3099
20.4.2.3 anchor (Anchor for Floating DrawingML Object) 3099
20.4.2.4 cNvGraphicFramePr (Common DrawingML Non-Visual Properties) 3106
20.4.2.5 docPr (Drawing Object Non-Visual Properties) 3107
20.4.2.6 effectExtent (Object Extents Including Effects) 3109
20.4.2.7 extent (Drawing Object Size) 3112
20.4.2.8 inline (Inline DrawingML Object) 3113
20.4.2.9 lineTo (Wrapping Polygon Line End Position) 3116
20.4.2.10 positionH (Horizontal Positioning) 3118
20.4.2.11 positionV (Vertical Positioning) 3119
20.4.2.12 posOffset (Absolute Position Offset) 3120
20.4.2.13 simplePos (Simple Positioning Coordinates) 3120
20.4.2.14 start (Wrapping Polygon Start) 3121
20.4.2.15 wrapNone (No Text Wrapping) 3122
20.4.2.16 wrapPolygon (Wrapping Polygon) 3123
20.4.2.17 wrapSquare (Square Wrapping) 3124
20.4.2.18 wrapThrough (Through Wrapping) 3127
20.4.2.19 wrapTight (Tight Wrapping) 3130
20.4.2.20 wrapTopAndBottom (Top and Bottom Wrapping) 3133
20.4.2.21 bg (Background Formatting) 3134
20.4.2.22 bodyPr (Body Properties) 3134
20.4.2.23 cNvCnPr (Non-Visual Connector Shape Drawing Properties) 3137
20.4.2.24 cNvContentPartPr (Non-Visual Content Part Drawing Properties) 3137
20.4.2.25 cNvFrPr (Non-Visual Graphic Frame Drawing Properties) 3138
20.4.2.26 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties) 3138
20.4.2.27 cNvPr (Non-Visual Drawing Properties) 3138
20.4.2.28 cNvSpPr (Non-Visual Drawing Properties for a Shape) 3140
20.4.2.29 contentPart (Content Part) 3140
20.4.2.30 extLst (Extension List) 3142
20.4.2.31 graphicFrame (Graphical object container) 3142
20.4.2.32 grpSp (Group Shape) 3142
20.4.2.33 grpSpPr (Group Shape Properties) 3142
20.4.2.34 linkedTxbx (Textual contents of shape) 3143
20.4.2.35 spPr (Shape Properties) 3143
20.4.2.36 style (Shape Style) 3143
20.4.2.37 txbx (Textual contents of shape) 3144
20.4.2.38 txbxContent (Rich Text Box Content Container) 3144
20.4.2.39 wgp (WordprocessingML Shape Group) 3144

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.4.2.40 whole (Whole E2O Formatting) 3144
20.4.2.41 wpc (WordprocessingML Drawing Canvas) 3144
20.4.2.42 wsp (WordprocessingML Shape) 3145
20.4.2.43 xfrm (2D Transform for Graphic Frames) 3145

20.4.3 Simple Types 3146

20.4.3.1 ST_AlignH (Relative Horizontal Alignment Positions) 3146
20.4.3.2 ST_AlignV (Vertical Alignment Definition) 3147
20.4.3.3 ST_PositionOffset (Absolute Position Offset Value) 3148
20.4.3.4 ST_RelFromH (Horizontal Relative Positioning) 3149
20.4.3.5 ST_RelFromV (Vertical Relative Positioning) 3150
20.4.3.6 ST_WrapDistance (Distance from Text) 3151
20.4.3.7 ST_WrapText (Text Wrapping Location) 3151

End of informative text.

20.4.2 Elements

The following elements define the contents of the WordprocessingML Drawing namespace:

20.4.2.1 align (Relative Horizontal Alignment)

This element specifies how a DrawingML object shall be horizontally aligned relative to the horizontal alignment base defined by the parent element. Once an alignment base is defined, this element shall determine how the DrawingML object shall be aligned relative to that location.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>left</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of `left` specifies that for the horizontal positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the left edge of the page. *end example*]

The possible values for this element are defined by the ST_AlignH simple type (§20.4.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_AlignH) is located in §A.4.4. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.4.2.2 align (Relative Vertical Alignment)

This element specifies how a DrawingML object shall be vertically aligned relative to the vertical alignment base defined by the parent element. Once an alignment base is defined, this element shall determine how the DrawingML object shall be aligned relative to that location.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>top</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of top specifies that for the vertical positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the top edge of the page. *end example*]

The possible values for this element are defined by the ST_AlignV simple type (§20.4.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (ST_AlignV) is located in §A.4.4. *end note*]

20.4.2.3 anchor (Anchor for Floating DrawingML Object)

This element specifies that the DrawingML object located at this position in the document is a floating object. Within a WordprocessingML document, drawing objects can exist in two states:

- *Inline* - The drawing object is in line with the text, and affects the line height and layout of its line (like a character glyph of similar size).
- *Floating* - The drawing object is anchored within the text, but can be absolutely positioned in the document relative to the page.

When this element encapsulates the DrawingML object's information, then all child elements shall dictate the positioning of this object as a floating object on the page.

[*Example*: Consider a WordprocessingML document where the anchor for a floating DrawingML object must be the first piece of run content within a paragraph. That paragraph's content would be specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<w:p>
  <w:r>
    <w:drawing>
      <wp:anchor … >
        …
      </wp:anchor>
    </w:drawing>
  </w:r>
</w:p>
```

The anchor element, when present as the child element of the drawing element, specifies that this DrawingML object must be positioned as a floating object based on the values of its child elements. *end example*]

Attributes	Description

allowOverlap (Allow Objects to Overlap)	Specifies whether a DrawingML object which intersects another DrawingML object at display time is allowed to overlap the contents of the other DrawingML object. If a DrawingML object cannot overlap other DrawingML object, it shall be repositioned when displayed to prevent this overlap as needed.
	
	[*Example*: Consider a document with two DrawingML objects which are allowed to overlap each other. This would be specified as follows within each object's anchor markup:
	
	`<wp:anchor allowOverlap="true" … >`
	`…`
	`</wp:anchor>`
	
	The allowOverlap attribute has a value of `true`, which specifies that this object must be allowed to overlap other objects when it is displayed on the document. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

behindDoc (Display Behind Document Text)	Specifies whether this floating DrawingML object is displayed behind the text of the document when the document is displayed. When a DrawingML object is displayed within a WordprocessingML document, that object can intersect with text in the document. This attribute shall determine whether the text or the object is rendered on top in case of overlapping.
	
	[*Example*: Consider a floating DrawingML object which must be displayed above any text which it intersects within the document's content. This setting would be specified as follows:
	
	`<wp:anchor behindDoc="false" … >`
	`…`
	`</wp:anchor>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The behindDoc attribute has a value of `false`, which specifies that the DrawingML object is displayed above the text of the document in z-order. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distB="457200" … >`
	`…`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<wp:anchor distL="228600" … > … </wp:anchor> The distL attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its right edge and the nearest text. This setting would be specified as follows: <wp:anchor distR="228600" … > … </wp:anchor> The distR attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

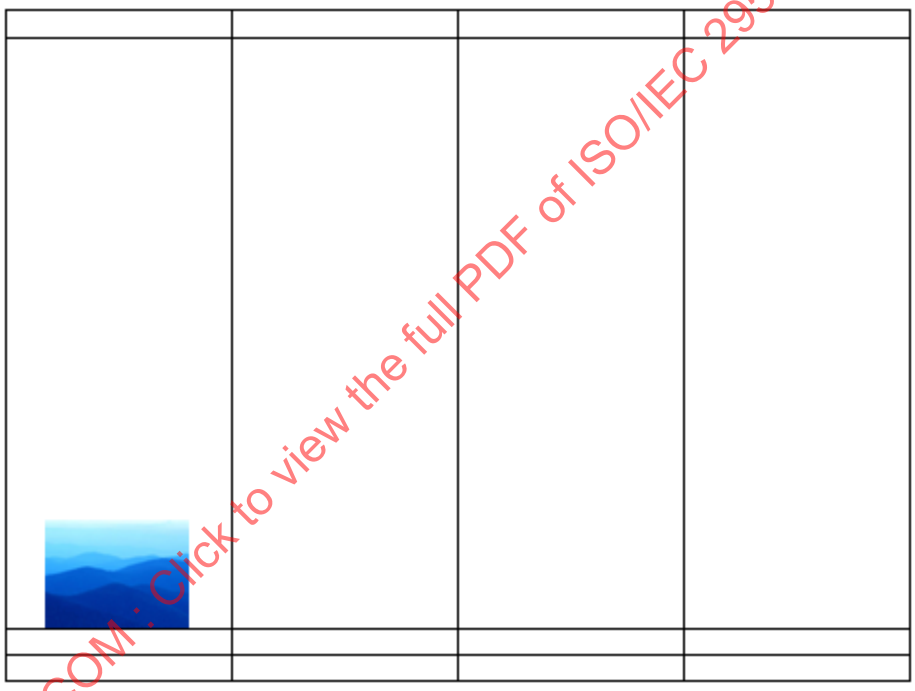
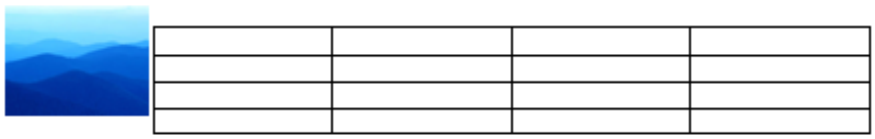
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distT="457200" … >`
	`…`
	`</wp:anchor>`
	
	The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

hidden (Hidden)	Specifies whether this floating DrawingML object is displayed. When a DrawingML object is displayed within a WordprocessingML document, that object can be hidden (i.e. present, but not visible). This attribute shall determine whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*]
	
	If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e. not hidden).
	
	[*Example*: Consider a floating DrawingML object which must be hidden within the document's content. This setting would be specified as follows:
	
	`<wp:anchor hidden="true" … >`
	`…`
	`</wp:anchor>`
	
	The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

layoutInCell (Layout In Table Cell)	Specifies how this DrawingML object behaves when its anchor is located in a table cell; and its specified position would cause it to intersect with a table cell displayed in the document. That behavior shall be as follows: • When this attribute has a value of `true`, then the object shall be positioned within the existing table cell, causing the cell to be resized as needed. This means that all positioning shall be relative to the cell and not the line on which the table appears. • When this attribute has a value of false, then the object shall be positioned as specified, but the table shall be resized and/or relocated within the document as needed to accommodate the object. This means that all positioning shall be

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

	relative to the line on which the table appears and not the cell in which the anchor is present.
	
	[*Example*: Consider a DrawingML picture which must be displayed in the center of the document. If the object is contained within a table and is defined as follows:
	
	`<wp:anchor layoutInCell="true" … >`
	`…`
	`</wp:anchor>`
	
	The layoutInCell attribute has a value of `true`, which specifies that the object can be placed within the cell if needed, for example:
	
	 
	
	If the layoutInCell attribute was now set to `false`, the object must be laid out outside of the cell, causing the table to be repositioned:
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

locked (Lock	Specifies that the anchor location for this object shall not be modified at runtime when

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

Anchor)	an application edits the contents of this document. [*Guidance*: An application might have automatic behaviors which reposition the anchor for a DrawingML object based on user interaction - for example, moving it from one page to another as needed. This element must tell applications not to perform any such behaviors. *end guidance*]
	
	[*Example*: Consider a floating DrawingML object which must have its anchor locked at the current location. This setting would be specified as follows:
	
	`<wp:anchor locked="true" … >`
	`…`
	`</wp:anchor>`
	
	The locked attribute has a value of `true`, which specifies that the DrawingML object's current anchor location must not be changed by applications editing this content. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

relativeHeight (Relative Z-Ordering Position)	Specifies the relative Z-ordering of all DrawingML objects in this document. Each floating DrawingML object shall have a Z-ordering value, which determines which object is displayed when any two objects intersect. Higher values shall indicate higher Z-order; lower values shall indicate lower Z-order.
	
	This attribute shall only indicate the Z-order with respect to other objects in the document which have an identical behindDoc attribute value. All objects with a behindDoc value of `false` shall be displayed above elements with a value of `true`.
	
	[*Example*: Consider two floating DrawingML objects as follows:
	
	`<wp:anchor relativeHeight="5" … >`
	`…`
	`</wp:anchor>`
	`…`
	
	`<wp:anchor relativeHeight="8" … >`
	`…`
	`</wp:anchor>`
	
	The relativeHeight attribute of the second object is 8, which specifies that the second DrawingML object must be at a higher Z-order than the first and must be displayed whenever the two overlap. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

simplePos (Page	Specifies that this object shall be positioned using the positioning information in the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Positioning)	simplePos child element (§20.4.2.13). This positioning, when specified, positions the object on the page by placing its top left point at the x-y coordinates specified by that element. If this element is omitted, then this object shall not use the simple positioning information in the simplePos element, even when present. [*Example*: Consider a floating DrawingML object which must be positioned at the top left corner of the page using simple positioning. This setting would be specified as follows: `<wp:anchor simplePos="true" … > <wp:simplePos x="0" y="0" /> … </wp:anchor>` The simplePos attribute has a value of `true`, which specifies that the DrawingML object's current position must be dictated by the simplePos element, and hence placed at 0,0. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Anchor) is located in §A.4.4. *end note*]

20.4.2.4 cNvGraphicFramePr (Common DrawingML Non-Visual Properties)

This element specifies common non-visual DrawingML object properties for the parent DrawingML object. These properties are specified as child elements of this element.

[*Example*: Consider a DrawingML object in a WordprocessingML document defined as follows:

```
<wp:inline>
  …
  <wp:cNvGraphicFramePr>
    <a:graphicFrameLocks … />
  </wp:cNvGraphicFramePr>
</wp:inline>
```

The cNvGraphicFramePr element contains a set of common non-visual properties as defined by DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.4.2.5 docPr (Drawing Object Non-Visual Properties)

This element specifies non-visual object properties for the parent DrawingML object. These properties are specified as child elements of this element.

[*Example*: Consider a DrawingML object in a WordprocessingML document defined as follows:

```
<wp:inline>
  …
  <wp:docPr id="1" name="Example Object">
    <a:hlinkClick … />
    <a:hlinkHover … />
  </wp:docPr>
</wp:inline>
```

The docPr element contains a set of common non-visual properties for this object. *end example*]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text which can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM: Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

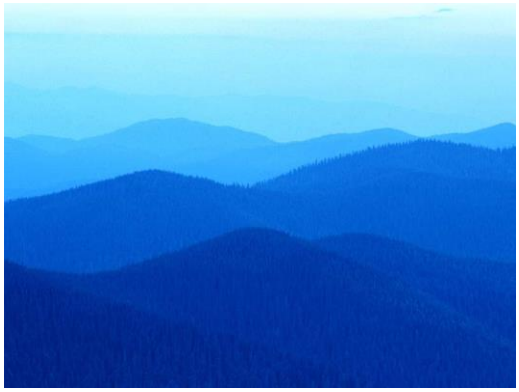
[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.4.2.6 effectExtent (Object Extents Including Effects)

This element specifies the additional extent which shall be added to each edge of the image (top, bottom, left, right) in order to compensate for any drawing effects applied to the DrawingML object.

The extent element (§20.4.2.7) specifies the size of the actual DrawingML object; however, an object can have effects applied which change its overall size [*Example*: A reflection and/or shadow effect. *end example*]. The additional size for each edge of the shape shall be stored on this element, and used to calculate the appropriate wrapping for wrap types without a wrapping polygon and the appropriate line height for inline objects.

[*Example*: Consider the following DrawingML image:



This object has no effects, and hence would have the following effect extents:

```
<wp:effectExtents b="0" t="0" l="0" r="0" />
```

However, if a shadow effect was applied which added effects to the right of the image:

Then the additional extent the right side would be specified in the r attribute on this element:

```
<wp:effectExtents b="0" t="0" l="0" r="695325" />
```

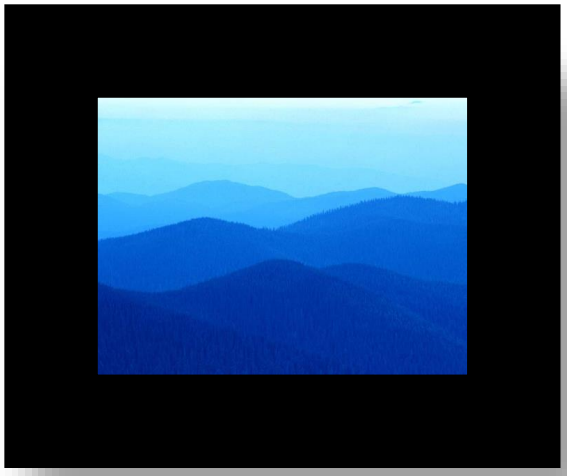
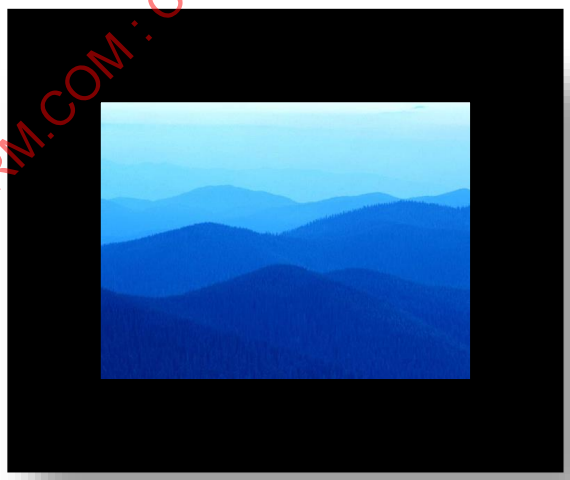
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The r attribute has a value of 695325, specifying that that 695325 EMUs must be added to the right side of the image. *end example*]

Attributes	Description
b (Additional Extent on Bottom Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image: This image has an effect on all four sides, resulting in the following markup: `<wp:effectExtent l="504825" t="447675" r="771525" b="809625" />` The b attribute value of 809625 specifies that 809625 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
l (Additional Extent on Left Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	 This image has an effect on all four sides, resulting in the following markup: `<wp:effectExtent l="504825" t="447675" r="771525" b="809625" />` The l attribute value of 504825 specifies that 504825 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
r (Additional Extent on Right Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image:  This image has an effect on all four sides, resulting in the following markup:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<wp:effectExtent l="504825" t="447675" r="771525" b="809625" /> The r attribute value of 771525 specifies that 771525 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
t (Additional Extent on Top Edge)	Specifies the additional length, in EMUs, which shall be added to the bottom edge of the DrawingML object to determine its actual bottom edge including effects. [*Example*: Consider the following DrawingML image: This image has an effect on all four sides, resulting in the following markup: <wp:effectExtent l="504825" t="447675" r="771525" b="809625" /> The t attribute value of 447675 specifies that 447675 additional EMUs must be added to the bottom of the image to compensate for the effects on the image. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_EffectExtent) is located in §A.4.4. *end note*]

20.4.2.7 extent (Drawing Object Size)

This element specifies the extents of the parent DrawingML object within the document (i.e. its final height and width).

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a DrawingML picture which is present in a WordprocessingML document and has an equal height and width. This object would be specified as follows:

```
<wp:anchor relativeHeight="10" allowOverlap="true">
  …
  <wp:extent cx="1828800" cy="1828800"/>
  …
</wp:anchor>
```

The extent element specifies via its attributes that this object has a height and width of 1828800 EMUs (English Metric Units). *end example*]

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.4.2.8 inline (Inline DrawingML Object)

This element specifies that the DrawingML object located at this position in the document is an inline object. Within a WordprocessingML document, drawing objects can exist in two states:

©ISO/IEC 2016 – All rights reserved

- *Inline* - The drawing object is in line with the text, and affects the line height and layout of its line (like a character glyph of similar size).
- *Floating* - The drawing object is anchored within the text, but can be absolutely positioned in the document relative to the page.

When this element encapsulates the DrawingML object's information, then all child elements shall dictate the positioning of this object in line with text.

[*Example*: Consider a WordprocessingML document where an inline DrawingML object must be the first piece of run content within a paragraph. That paragraph's content would be specified as follows:

```
<w:p>
  <w:r>
    <w:drawing>
      <wp:inline>
        …
      </wp:inline>
    </w:drawing>
  </w:r>
</w:p>
```

The inline element, when present as the child element of the drawing element, specifies that this DrawingML object must be positioned in line with the text of this paragraph, modifying line heights, etc. as necessary. *end example*]

Attributes	Description

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distB="457200" … >`
	`…`
	`</wp:anchor>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The distB attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored.
	
	[*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor distL="228600" … >`
	`…`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-quarter of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<wp:anchor distR="228600" … > … </wp:anchor> The distR attribute specifies that the padding distance must be 228600 EMUs or one-quarter of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). If this object is an inline object (i.e. has a parent element of inline), then this value shall not have any effect when displaying the object in line with text, but can be maintained and used if the object is subsequently changed to floating. If the wrapping element [*Example*: wrapThrough or wrapSquare *end example*] present as a child element also has a distance from text, then this value shall be ignored. [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: <wp:anchor distT="457200" … > … </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Inline) is located in §A.4.4. *end note*]

20.4.2.9 lineTo (Wrapping Polygon Line End Position)

This element specifies a single point on the wrapping polygon for a DrawingML object. This point shall be the termination of the edge of the wrapping polygon started by the previous start or lineTo element in document order, and shall be the origin of the next edge on the same polygon.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The attributes on this element shall dictate the position of the point relative to the upper-left corner of the actual object.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The lineTo element defines each point of the wrapping polygon (in this case, the four points of the wrapping square). *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.4.2.10 positionH (Horizontal Positioning)

This element specifies the horizontal positioning of a floating DrawingML object within a WordprocessingML document. This positioning is specified in two parts:

- Positioning Base - The relativeFrom attribute on this element specifies the part of the document from which the positioning shall be calculated.
- Positioning - The child element of this element (align or posOffset) specifies how the object is positioned relative to that base.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionH>
  <wp:positionV relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionV>
</wp:anchor>
```

The positionH element specifies that the object is horizontally positioned relative to the margin via the relativeFrom attribute; and that the alignment relative to the margin is centered via the align element. *end example*]

Attributes	Description

relativeFrom (Horizontal Position Relative Base)	Specifies the base to which the relative horizontal positioning of this object shall be calculated.
	
	[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:
	
	`<wp:anchor … >`
	`  <wp:positionH relativeFrom="page">`
	`  <wp:align>center</wp:align>`
	`  </wp:positionH>`
	`  …`
	`</wp:anchor>`
	
	The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]
	
	The possible values for this attribute are defined by the ST_RelFromH simple type (§20.4.3.4).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_PosH) is located in §A.4.4. *end note*]

20.4.2.11 positionV (Vertical Positioning)

This element specifies the vertical positioning of a floating DrawingML object within a WordprocessingML document. This positioning is specified in two parts:

- Positioning Base - The relativeFrom attribute on this element specifies the part of the document from which the positioning shall be calculated.
- Positioning - The child element of this element (align or posOffset) specifies how the object is positioned relative to that base.

[*Example*: Consider a DrawingML picture which must be displayed in the center of the printed page on which it appears, modifying the flow of text as necessary. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionH>
  <wp:positionV relativeFrom="margin">
    <wp:align>center</wp:align>
  </wp:positionV>
</wp:anchor>
```

The positionV element specifies that the object is vertically positioned relative to the margin via the relativeFrom attribute; and that the alignment relative to the margin is centered via the align element. *end example*]

Attributes	Description

relativeFrom (Vertical Position Relative Base)	Specifies the base to which the relative vertical positioning of this object shall be calculated.
	
	[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page margins. This object would be specified as follows:
	
	`<wp:anchor … >`
	`…`
	`<wp:positionV relativeFrom="margin">`
	`<wp:align>bottom</wp:align>`
	`</wp:positionV>`
	`</wp:anchor>`
	
	The relativeFrom attribute specifies that the object is horizontally positioned relative to the margin. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_RelFromV simple type (§20.4.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PosV) is located in §A.4.4. *end note*]

20.4.2.12 posOffset (Absolute Position Offset)

This element specifies an absolute measurement for the positioning of a floating DrawingML object within a WordprocessingML document. This measurement shall be calculated relative to the top left edge of the positioning base specified by the parent element's relativeFrom attribute.

[*Example*: Consider a DrawingML picture which must be displayed one inch from the top of the page, and one-half of an inch from the left edge of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:posOffset>914400</wp:posOffset>
  </wp:positionH>
  <wp:positionV relativeFrom="page">
    <wp:posOffset>457200</wp:posOffset>
  </wp:positionV>
</wp:anchor>
```

The posOffset element specifies the absolute positioning of the object relative to the top-left edge of the page in EMUs. *end example*]

The possible values for this element are defined by the ST_PositionOffset simple type (§20.4.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_PositionOffset) is located in §A.4.4. *end note*]

20.4.2.13 simplePos (Simple Positioning Coordinates)

This element specifies the coordinates at which a DrawingML object shall be positioned relative to the top-left edge of its page, when the simplePos attribute is specified on the anchor element (§20.4.2.3).

[*Example*: Consider a floating DrawingML object which must be positioned at the top left corner of the page using simple positioning. This setting would be specified as follows:

```
<wp:anchor simplePos="true" … >
  <wp:simplePos x="0" y="0" />
  …
</wp:anchor>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The simplePos attribute has a value of `true`, which specifies that the DrawingML object's current position must be dictated by the simplePos element, and hence placed at `0,0`. *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.4.2.14 start (Wrapping Polygon Start)

This element specifies the starting point on the wrapping polygon for a DrawingML object. This point shall be the start and termination of the wrapping polygon for the parent object.

The attributes on this element shall dictate the position of the point relative to the upper-left corner of the actual object.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The start element defines the start and end of the wrapping polygon (in this case, the four points of the wrapping square). *end example*]

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of 0. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of 100. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.4.2.15 wrapNone (No Text Wrapping)

This element specifies that the parent DrawingML object shall not cause any text wrapping within the contents of the host WordprocessingML document based on its display location. In effect, this setting shall place the object in one of two locations:

 ©ISO/IEC 2016 – All rights reserved

- If the behindDoc attribute on the parent element is `true`, then the object shall be positioned behind the text as it is normally displayed.
- If the behindDoc attribute on the parent element is `false`, then the object shall be positioned in front of the text as it is normally displayed.

[*Example*: Consider a DrawingML picture which must be displayed in front of any text on the page. This object would be specified as follows:

```
<wp:anchor relativeHeight="10" behindDoc="false">
  …
  <wp:wrapNone/>
</wp:anchor>
```

The wrapNone element specifies that the DrawingML object must not cause any text wrapping, and since the behindDoc attribute is `false`, the object must be displayed in front of the text of the document. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapNone) is located in §A.4.4. *end note*]

20.4.2.16 wrapPolygon (Wrapping Polygon)

This element specifies the wrapping polygon which shall be used to determine the extents to which text can wrap around the specified object in the document. This polygon shall be defined by the following:

- The start element defines the coordinates of the origin of the wrap polygon
- Two or more lineTo elements define the point of the wrap polygon

If the set of child elements does not result in a closed polygon (the last lineTo element does not return to the position specified by the start element), then a single additional line shall be inferred as needed to close the wrapping polygon.

[*Example*: Consider the following basic wrapping polygon for a DrawingML object:

```
<wp:wrapPolygon>
  <wp:start x="0" y="0" />
  <wp:lineTo x="0" y="100" />
  <wp:lineTo x="100" y="100" />
  <wp:lineTo x="100" y="0" />
  <wp:lineTo x="0" y="0" />
</wp:wrapPolygon>
```

The wrapPolygon element defines the object's text wrapping polygon (in this case, the four points of a square). *end example*]

Attributes	Description
edited (Wrapping	Specifies that the wrap points for the wrapping polygon have been edited, and the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Points Modified)	resulting extents shall be recalculated to compensate when the document is next opened.
	
	[*Example*: Consider the following basic wrapping polygon for a DrawingML object:
	
	`<wp:wrapPolygon edited="true">`
	`<wp:start x="0" y="0" />`
	`<wp:lineTo x="0" y="100" />`
	`<wp:lineTo x="50" y="50" />`
	`<wp:lineTo x="0" y="0" />`
	`</wp:wrapPolygon>`
	
	The edited attribute specifies that these wrap points have been changed since the document was last rendered. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapPath) is located in §A.4.4. *end note*]

20.4.2.17 wrapSquare (Square Wrapping)

This element specifies that text shall wrap around a virtual rectangle bounding this object. The bounds of the wrapping rectangle shall be dictated by the extents including the addition of the effectExtent element as a child of this element (if present) or the effectExtent present on the parent element.

[*Example*: Consider a DrawingML object using square wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapSquare wrapText="bothSides" />
</wp:anchor>
```

The wrapSquare element specifies that text must wrap around both sides of a rectangle around this object which includes its effect extents. *end example*]

Attributes	Description
distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distB="457200" … />`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapSquare distR="457200" … />`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</wp:anchor> The distR attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
distT (Distance From Text (Top))	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: <wp:anchor … > … <wp:wrapSquare distT="457200" … /> </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).
wrapText (Text Wrapping Location)	Specifies how text shall wrap around the object's left and right sides. [*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting would be specified as follows: <wp:anchor … > … <wp:wrapSquare wrapText="left" … /> </wp:anchor> The wrapText attribute value of left specifies that text must only wrap around the let side of the object. *end example*] The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapSquare) is located in §A.4.4. *end note*]

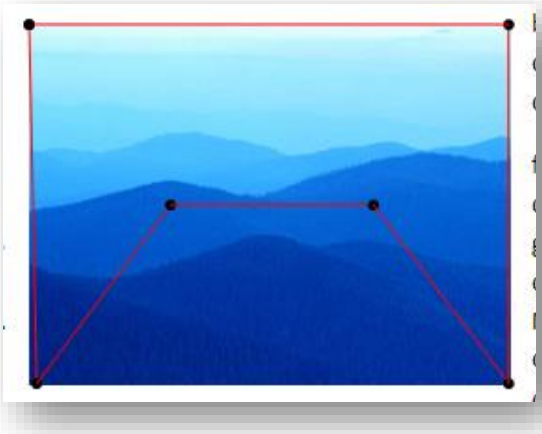
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.4.2.18 wrapThrough (Through Wrapping)

This element specifies that text shall wrap around the wrapping polygon bounding this object as defined by the child wrapPolygon element. When this element specifies a wrapping polygon, it shall allow text to wrap within the object's maximum left and right extents.

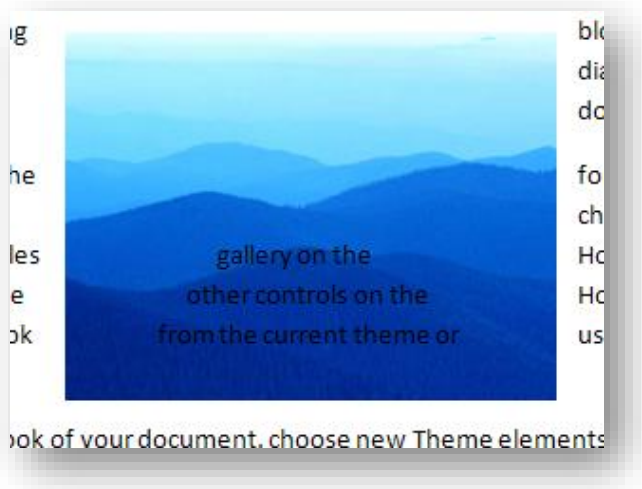
[*Example*: Consider an object with the following wrap points:



If this object uses tight wrapping, then text cannot be placed within the maximum left and right extents of the wrap polygon at any location:

However, with through wrapping:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016



end example]

[*Example*: Consider a DrawingML object using through wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapThrough wrapText="bothSides">
    …
  </wp:wrapThrough>
</wp:anchor>
```

The wrapThrough element specifies that text must wrap through this object as defined by its wrap polygon. *end example*]

Attributes	Description

distL (Distance From Text on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapThrough distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`…`
	`<wp:wrapThrough distR="457200" … />`
	`</wp:anchor>`
	
	The distR attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

wrapText (Text Wrapping Location)	Specifies how text shall wrap around the object's left and right sides.
	
	[*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`…`
	`<wp:wrapThrough wrapText="left" … />`
	`</wp:anchor>`
	
	The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

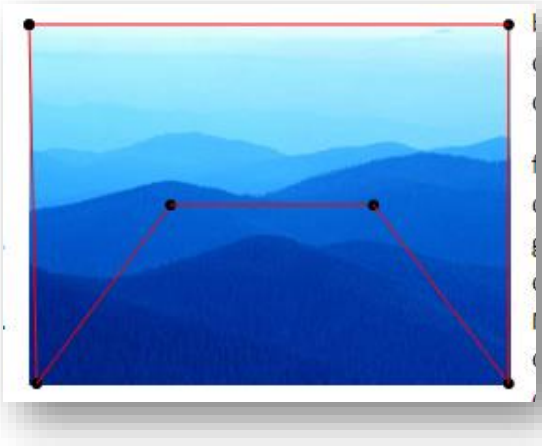
[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapThrough) is located in §A.4.4. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.4.2.19 wrapTight (Tight Wrapping)

This element specifies that text shall wrap around the wrapping polygon bounding this object as defined by the child wrapPolygon element. When this element specifies a wrapping polygon, it shall not allow text to wrap within the object's maximum left and right extents.

[*Example*: Consider an object with the following wrap points:

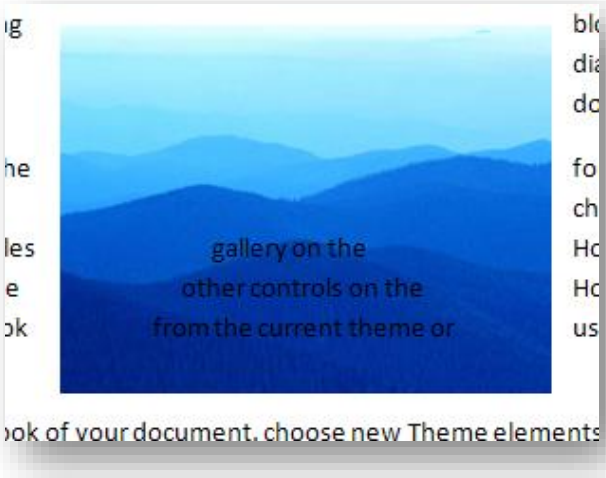


If this object uses tight wrapping, then text cannot be placed within the maximum left and right extents of the wrap polygon at any location:

However, with through wrapping:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



end example]

[*Example*: Consider a DrawingML object using tight wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapTight wrapText="bothSides">
    …
  </wp:wrapTight>
</wp:anchor>
```

The wrapTight element specifies that text must wrap through this object as defined by its wrap polygon. *end example*]

Attributes	Description

distL (Distance From Test on Left Edge)	Specifies the minimum distance which shall be maintained between the left edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight distL="457200" … />`
	`</wp:anchor>`
	
	The distL attribute specifies that the padding distance must be 457200 EMUs or one-half

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distR (Distance From Text on Right Edge)	Specifies the minimum distance which shall be maintained between the right edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its right edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight distR="457200" … />`
	`</wp:anchor>`
	
	The distR attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

wrapText (Text Wrapping Location)	Specifies how text shall wrap around the object's left and right sides.
	
	[*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTight wrapText="left" … />`
	`</wp:anchor>`
	
	The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapText simple type (§20.4.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapTight) is located in §A.4.4. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.4.2.20 wrapTopAndBottom (Top and Bottom Wrapping)

This element specifies that text shall wrap around the top and bottom of this object, but not its left or right edges.

[*Example*: Consider a DrawingML object using top and bottom wrapping and defined as follows:

```
<wp:anchor … >
  …
  <wp:wrapTopAndBottom />
</wp:anchor>
```

The wrapTopAndBottom element specifies that text must wrap around neither side of this object. *end example*]

Attributes	Description

distB (Distance From Text on Bottom Edge)	Specifies the minimum distance which shall be maintained between the bottom edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents.
	
	The distance shall be measured in EMUs (English Metric Units).
	
	[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its bottom edge and the nearest text. This setting would be specified as follows:
	
	`<wp:anchor … >`
	`  …`
	`  <wp:wrapTopAndBottom distB="457200" … />`
	`</wp:anchor>`
	
	The distB attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]
	
	The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

distT (Distance From Text on Top Edge)	Specifies the minimum distance which shall be maintained between the top edge of this drawing object and any subsequent text within the document when this graphical object is displayed within the document's contents. The distance shall be measured in EMUs (English Metric Units). [*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its top edge and the nearest text. This setting would be specified as follows: `<wp:anchor … >`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… <wp:wrapTopAndBottom distT="457200" … /> </wp:anchor> The distT attribute specifies that the padding distance must be 457200 EMUs or one-half of an inch. *end example*] The possible values for this attribute are defined by the ST_WrapDistance simple type (§20.4.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_WrapTopBottom) is located in §A.4.4. *end note*]

20.4.2.21 bg (Background Formatting)

This element defines formatting that can be applied to the background shape of the document. The background shape can hold formatting options just as a normal shape can hold within DrawingML.

20.4.2.22 bodyPr (Body Properties)

This element defines the body properties for the text body within a shape.

Attributes	Description
anchor (Anchor)	Specifies the anchoring position of the txBody within the shape. If this attribute is omitted, then a value of t, or top is implied. The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.60).
anchorCtr (Anchor Center)	Specifies the centering of the text box. The way it works fundamentally is to determine the smallest possible "bounds box" for the text and then to center that "bounds box" accordingly. This is different than paragraph alignment, which aligns the text within the "bounds box" for the text. This flag is compatible with all of the different kinds of anchoring. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bIns (Bottom Inset)	Specifies the bottom inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, a value of 45720 or 0.05 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
compatLnSpc (Compatible Line	Specifies that the line spacing for this text body is decided in a simplistic manner using the font scene. If this attribute is omitted, a value of 0 or false is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Spacing)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
forceAA (Force Anti-Alias)	Forces the text to be rendered anti-aliased regardless of the font size. Certain fonts can appear grainy around their edges unless they are anti-aliased. Therefore this attribute allows for the specifying of which bodies of text should always be anti-aliased and which ones should not. If this attribute is omitted, then a value of `0` or `false` is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fromWordArt (From WordArt)	Specifies that text within this textbox is converted text from a WordArt object. This is more of a backwards compatibility attribute that is useful to the application from a tracking perspective. WordArt was the former way to apply text effects and therefore this attribute is useful in document conversion scenarios. If this attribute is omitted, then a value of `0` or `false` is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
horzOverflow (Text Horizontal Overflow)	Determines whether the text can flow out of the bounding box horizontally. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.69).
lIns (Left Inset)	Specifies the left inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
numCol (Number of Columns)	Specifies the number of columns of text in the bounding rectangle. When applied to a text run this property takes the width of the bounding box for the text and divides it by the number of columns specified. These columns are then treated as overflow containers in that when the previous column has been filled with text the next column acts as the repository for additional text. When all columns have been filled and text still remains then the overflow properties set for this text body are used and the text is reflowed to make room for additional text. If this attribute is omitted, then a value of 1 is implied. The possible values for this attribute are defined by the ST_TextColumnCount simple type (§20.1.10.65).
rIns (Right Inset)	Specifies the right inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
rot (Rotation)	Specifies the rotation that is being applied to the text within the bounding box. If it not specified, the rotation of the accompanying shape is used. If it is specified, then this is applied independently from the shape. That is the shape can have a rotation applied in addition to the text itself having a rotation applied to it. If this attribute is omitted, then a value of 0, is implied. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).
rtlCol (Columns Right-To-Left)	Specifies whether columns are used in a right-to-left or left-to-right order. The usage of this attribute only sets the column order that is used to determine which column overflow text should go to next. If this attribute is omitted, then a value of `0` or `false`is implied in which case text starts in the leftmost column and flow to the right. [*Note*: This attribute in no way determines the direction of text but merely the direction in which multiple columns are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spcCol (Space Between Columns)	Specifies the space between text columns in the text area. This should only apply when there is more than 1 column present. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).
spcFirstLastPara (Paragraph Spacing)	Specifies whether the before and after paragraph spacing defined by the user is to be respected. While the spacing between paragraphs is helpful, it is additionally useful to be able to set a flag as to whether this spacing is to be followed at the edges of the text body, in other words the first and last paragraphs in the text body. More precisely since this is a text body level property it should only effect the before paragraph spacing of the first paragraph and the after paragraph spacing of the last paragraph for a given text body. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tIns (Top Inset)	Specifies the top inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 45720 or 0.05 inches is implied. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
upright (Text	Specifies whether text should remain upright, regardless of the transform applied to it

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Upright)	and the accompanying shape transform. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vert (Vertical Text)	Determines if the text within the given text body should be displayed vertically. If this attribute is omitted, then a value of horz, or no vertical text is implied. The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.83).
vertOverflow (Text Vertical Overflow)	Determines whether the text can flow out of the bounding box vertically. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. The possible values for this attribute are defined by the ST_TextVertOverflowType simple type (§20.1.10.84).
wrap (Text Wrapping Type)	Specifies the wrapping options to be used for this text body. If this attribute is omitted, then a value of square is implied which wraps the text using the bounding text box. The possible values for this attribute are defined by the ST_TextWrappingType simple type (§20.1.10.85).

20.4.2.23 cNvCnPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual drawing properties specific to a connector shape. This includes information specifying the shapes to which the connector shape is connected.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.4. *end note*]

20.4.2.24 cNvContentPartPr (Non-Visual Content Part Drawing Properties)

This element specifies the non-visual drawing properties for a content part. This allows for additional information that does not affect the appearance of the content part to be stored.

Attributes	Description
isComment (Is a Comment) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether the content part is a comment or an annotation. If `true`, it is a comment; otherwise, it is a general annotation. The default value for this attribute is `true`. [*Example*: Consider a WordprocessingDrawingML object defined as follows: `<… isComment="false">`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The content part shape does not represent a comment. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualContentPartProperties) is located in §A.4.1. *end note*]

20.4.2.25 cNvFrPr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.4.2.26 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering.

20.4.2.27 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following WordprocessingDrawingML:

```
<wsp>
  …
  <cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  …
</wsp>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications that do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<… descr="A picture of a bowl of fruit"> The descr attribute contains alternative text that can be used in place of the actual DrawingML object. *end example]* The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object that must be hidden within the document's content. This setting would be specified as follows: <… hidden="true" /> The hidden attribute has a value of true, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example]* The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: <… id="10" … > The id attribute has a value of 10, which is the unique identifier for this DrawingML object. *end example]* The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.or	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
g/ooxml/drawingml/main	`< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example]* The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example]* The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.4.2.28 cNvSpPr (Non-Visual Drawing Properties for a Shape)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with.

Attributes	Description
txBox (Text Box) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

20.4.2.29 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ECMA-376. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

Attributes	Description
bwMode (Black and White Mode)	Specifies how to interpret color information contained within a content part to achieve a color, black and white, or grayscale rendering of the content part. This attribute specifies only the rendering mode applied to the content part; it does not affect how the actual color information is persisted. The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).
id (Relationship to Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the relationship type required by the parent element: • http://purl.oclc.org/ooxml/officeDocument/customXml for the contentPart element • http://purl.oclc.org/ooxml/officeDocument/relationships/footer for the footerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/header for the headerReference element • http://purl.oclc.org/ooxml/officeDocument/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements • http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings for the printerSettings element • http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink for the longDesc or hyperlink element [*Example*: Consider an XML element which has the following id attribute: <… r:id="rId10" /> The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_WordprocessingContentPart) is located in §A.4.4. *end note*]

20.4.2.30 extLst (Extension List)

This element specifies an extension list, within which all future extensions are defined within ext elements.

The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the existing diagram syntax.

20.4.2.31 graphicFrame (Graphical object container)

This element specifies a container for a graphical object in WordprocessingML.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicFrame) is located in §A.4.5. *end note*]

20.4.2.32 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

20.4.2.33 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.4.2.34 linkedTxbx (Textual contents of shape)

This element specifies the textual contents of a shape that is not the first in the series of shapes for the same text box story.

Attributes	Description
id (ID)	Specifies the identity of the text box story begun by a txbx element. This value shall be unique across a document for each txbx element. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.
seq (sequence index)	Specifies the position of the owning shape in the given text box story. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LinkedTextboxInformation) is located in §A.4.4. *end note*]

20.4.2.35 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a shape. These properties include the shape fill, outline, geometry, effects, and 3D orientation.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

20.4.2.36 style (Shape Style)

This element specifies the style information for a shape. This is used to define a shape's appearance in terms of the preset styles defined by the style matrix for the theme.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.4.2.37 txbx (Textual contents of shape)

This element specifies the textual contents of a shape which is the first in the series of shapes for the same text box story. This element shall be present only in the CT_WordprocessingShape element that is the first in a series of CT_WordprocessingShape elements that refer to the same text box story.

Attributes	Description
id (ID)	Specifies the identity of the text box story begun by a txbx element. This value shall be unique across a document for each txbx element. The possible values for this attribute are defined by the W3C XML Schema unsignedShort datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextboxInfo) is located in §A.4.4. *end note*]

20.4.2.38 txbxContent (Rich Text Box Content Container)

This element specifies that its contents shall be any rich WordprocessingML content, and that this content is the rich contents of a drawing object defined using DrawingML syntax.

If this element contains within any of its contents any of the following content, then the document shall be considered non-conformant:

- References to other WordprocessingML document stories (comments, footnotes, endnotes)
- Vector Markup Language (VML)
- Additional txbxContent elements (as part of nested DrawingML objects)

20.4.2.39 wgp (WordprocessingML Shape Group)

This element specifies a shape group in WordprocessingML.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ WordprocessingGroup) is located in §A.4.5. *end note*]

20.4.2.40 whole (Whole E2O Formatting)

Formatting that applies to the entire diagram object, and not just the background, includes line and effect properties.

20.4.2.41 wpc (WordprocessingML Drawing Canvas)

This element specifies a drawing canvas in WordprocessingML. A drawing canvas is a logical grouping of shapes. [*Note*: A Drawing Canvas is typically used to allow grouping of shapes together for bulk operations. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_ WordprocessingCanvas) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.4.2.42 wsp (WordprocessingML Shape)

This element specifies a shape in WordprocessingML.

Attributes	Description
normalEastAsianFlow (East Asian Flow)	Specifies that the text flow of the text contents of the shape shall ignore the text flow value specified by the vert attribute of the bodyPr element. If this attribute is set to TRUE then the text flows in the manner specified by the value "tbV" for ST_TextDirection (§17.18.93). The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ WordprocessingShape) is located in §A.4.5. *end note*]

20.4.2.43 xfrm (2D Transform for Graphic Frames)

This element specifies a two dimensional transform for a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

20.4.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – WordprocessingML Drawing.

20.4.3.1 ST_AlignH (Relative Horizontal Alignment Positions)

This simple type contains the possible settings specifying how a DrawingML object can be horizontally aligned relative to the horizontal alignment base defined by the parent element.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>left</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of `left` specifies that for the horizontal positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the left edge of the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center Alignment)	Specifies that the object shall be centered with respect to the horizontal alignment base. [*Example*: Centered on the page. *end example*]
inside (Inside)	Specifies that the object shall be inside of the horizontal alignment base.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: Inside the outside margin. *end example*]
left (Left Alignment)	Specifies that the object shall be left aligned to the horizontal alignment base. [*Example*: Left aligned relative to the margins. *end example*]
outside (Outside)	Specifies that the object shall be outside of the horizontal alignment base. [*Example*: Outside the left margin. *end example*]
right (Right Alignment)	Specifies that the object shall be right aligned to the horizontal alignment base. [*Example*: Right aligned relative to the margins. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlignH) is located in §A.4.4. *end note*]

20.4.3.2 ST_AlignV (Vertical Alignment Definition)

This simple type contains the possible settings specifying how a DrawingML object can be vertically aligned relative to the vertical alignment base defined by the parent element.

[*Example*: Consider a picture in a WordprocessingML document which has been aligned relative to the edge of the page - the left of the page horizontally, and the top of the page vertically. This alignment would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>top</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The align element with a value of top specifies that for the vertical positioning defined by the parent element (in this case, positioning relative to the page), the picture must be aligned to the top edge of the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Bottom)	Specifies that the object shall be at the bottom of the vertical alignment base. [*Example*: Bottom of the page. *end example*]
center (Center Alignment)	Specifies that the object shall be centered with respect to the vertical alignment base. [*Example*: Centered on the page. *end example*]
inside (Inside)	Specifies that the object shall be inside of the horizontal alignment base. [*Example*: Inside the top margin. *end example*]
outside (Outside)	Specifies that the object shall be outside of the vertical alignment base. [*Example*: Outside the top margin. *end example*]
top (Top)	Specifies that the object shall be at the top of the vertical alignment base. [*Example*: Top of the page. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlignV) is located in §A.4.4. *end note*]

20.4.3.3 ST_PositionOffset (Absolute Position Offset Value)

This simple type represents a one dimensional distance which shall be used to offset an objet from its base positioning location stored in EMUs.

[*Example*: Consider a DrawingML picture which must be displayed one inch from the top of the page, and one-half of an inch from the left edge of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:posOffset>914400</wp:posOffset>
  </wp:positionH>
  <wp:positionV relativeFrom="page">
    <wp:posOffset>457200</wp:posOffset>
  </wp:positionV>
</wp:anchor>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The posOffset element specifies the absolute positioning of the object relative to the top-left edge of the page in EMUs. *end example*]

This simple type's contents are a restriction of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositionOffset) is located in §A.4.4. *end note*]

20.4.3.4 ST_RelFromH (Horizontal Relative Positioning)

This simple type specifies the possible values for the base from which the relative horizontal positioning of an object shall be calculated.

[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionH relativeFrom="page">
    <wp:align>center</wp:align>
  </wp:positionH>
  …
</wp:anchor>
```

The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
character (Character)	Specifies that the horizontal positioning shall be relative to the position of the anchor within its run content.
column (Column)	Specifies that the horizontal positioning shall be relative to the extents of the column which contains its anchor.
insideMargin (Inside Margin)	Specifies that the horizontal positioning shall be relative to the inside margin of the current page (the left margin on odd pages, right on even pages).
leftMargin (Left Margin)	Specifies that the horizontal positioning shall be relative to the left margin of the page.
margin (Page Margin)	Specifies that the horizontal positioning shall be relative to the page margins.
outsideMargin (Outside Margin)	Specifies that the horizontal positioning shall be relative to the outside margin of the current page (the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	right margin on odd pages, left on even pages).
page (Page Edge)	Specifies that the horizontal positioning shall be relative to the edge of the page.
rightMargin (Right Margin)	Specifies that the horizontal positioning shall be relative to the right margin of the page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RelFromH) is located in §A.4.4. *end note*]

20.4.3.5 ST_RelFromV (Vertical Relative Positioning)

This simple type specifies the possible values for the base from which the relative vertical positioning of an object shall be calculated.

[*Example*: Consider a DrawingML picture which must be displayed at the bottom center of the page. This object would be specified as follows:

```
<wp:anchor … >
  <wp:positionV relativeFrom="page">
    <wp:align>bottom</wp:align>
  </wp:positionV>
  …
</wp:anchor>
```

The relativeFrom attribute specifies that the object is horizontally positioned relative to the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottomMargin (Bottom Margin)	Specifies that the vertical positioning shall be relative to the bottom margin of the current page.
insideMargin (Inside Margin)	Specifies that the vertical positioning shall be relative to the inside margin of the current page.
line (Line)	Specifies that the vertical positioning shall be relative to the line containing the anchor character.
margin (Page Margin)	Specifies that the vertical positioning shall be relative to the page margins.
outsideMargin (Outside Margin)	Specifies that the vertical positioning shall be relative to the outside margin of the current page.
page (Page Edge)	Specifies that the vertical positioning shall be relative

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	to the edge of the page.
paragraph (Paragraph)	Specifies that the vertical positioning shall be relative to the paragraph which contains the drawing anchor.
topMargin (Top Margin)	Specifies that the vertical positioning shall be relative to the top margin of the current page.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RelFromV) is located in §A.4.4. *end note*]

20.4.3.6 ST_WrapDistance (Distance from Text)

This simple type represents a one dimensional distance which shall be used to offset an object from text, stored in EMUs.

[*Example*: Consider a floating DrawingML object which must have one-half of an inch of padding between its left edge and the nearest text. This setting would be specified as follows:

```
<wp:anchor … >
  …
  <wp:wrapThrough distL="457200" … />
</wp:anchor>
```

The distL attribute specifies that the padding distance must be `457200` EMUs or one-half of an inch. *end example*]

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapDistance) is located in §A.4.4. *end note*]

20.4.3.7 ST_WrapText (Text Wrapping Location)

This simple type specifies the possible settings for how text can wrap around the object's left and right sides.

[*Example*: Consider a floating DrawingML object which must allow text to wrap around its left side only. This setting would be specified as follows:

```
<wp:anchor … >
  …
  <wp:wrapTight wrapText="left" … />
</wp:anchor>
```

The wrapText attribute value of `left` specifies that text must only wrap around the let side of the object. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bothSides (Both Sides)	Specifies that text shall wrap around both sides of the object.
largest (Largest Side Only)	Specifies that text shall only wrap around the largest side of the object. If the object is positioned in the exact center of the page, the text shall wrap around the side on which text is first encountered: • If the first line of text intersecting the object is using left-to-right reading order, the text shall wrap to the object's left. • If the first line of text intersecting the object is using right-to-left reading order, the text shall wrap to the object's right.
left (Left Side Only)	Specifies that text shall only wrap around the left side of the object.
right (Right Side Only)	Specifies that text shall only wrap around the right side of the object.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_WrapText) is located in §A.4.4. *end note*]

20.5 DrawingML - SpreadsheetML Drawing

Within a SpreadsheetML document, it is possible to include graphical DrawingML objects:

- Pictures (§20.2)
- Locked Canvases (§20.3)
- Diagrams (§21.4)
- Charts (§21.2)

When these objects are present in a spreadsheet document, it is necessary to include information which specifies how the objects shall be positioned relative to the parent worksheet. [*Example*: Whether the object is anchored to a specific row, whether it resizes with cells, and so on. *end example*]

The SpreadsheetML Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a spreadsheet document.

[*Example*: Consider a DrawingML picture which must be anchored to a specific cell for its top left and bottom right corners, resizing as those cells are relocated. This object would be specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<xdr:twoCellAnchor>
  <xdr:from>
    …
  </xdr:from>
  <xdr:to>
    …
  </xdr:to>
  <xdr:graphicFrame>
    …
    <a:graphic>
      <a:graphicData
         uri="http://purl.oclc.org/ooxml/drawingml/diagram">
        <dgm:relIds xmlns:dgm="…" xmlns:r="…" r:dm="rId1" r:lo="rId2"
           r:qs="rId3" r:cs="rId4" />
      </a:graphicData>
    </a:graphic>
  </xdr:graphicFrame>
</xdr:twoCellAnchor>
```

The twoCellAnchor element (§20.5.2.33) specifies that this object anchored to the cells specified by the to (§20.5.2.32) and from (§20.5.2.15) elements. *end example*]

20.5.1 Table of Contents

This subclause is informative.

20.5.2 Elements 3154

20.5.2.1 absoluteAnchor (Absolute Anchor Shape Size) 3154
20.5.2.2 blipFill (Picture Fill) 3154
20.5.2.3 clientData (Client Data) 3157
20.5.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties) 3157
20.5.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties) 3157
20.5.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties) 3157
20.5.2.7 cNvPicPr (Non-Visual Picture Drawing Properties) 3158
20.5.2.8 cNvPr (Non-Visual Drawing Properties) 3159
20.5.2.9 cNvSpPr (Connection Non-Visual Shape Properties) 3161
20.5.2.10 col (Column)) 3161
20.5.2.11 colOff (Column Offset) 3162
20.5.2.12 contentPart (Content Part) 3162
20.5.2.13 cxnSp (Connection Shape) 3166
20.5.2.14 ext (Shape Extent) 3166
20.5.2.15 from (Starting Anchor Point) 3167
20.5.2.16 graphicFrame (Graphic Frame) 3168
20.5.2.17 grpSp (Group Shape) 3169
20.5.2.18 grpSpPr (Group Shape Properties) 3170
20.5.2.19 nvCxnSpPr (Non-Visual Properties for a Connection Shape) 3170

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.5.2.20 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame) 3170
20.5.2.21 nvGrpSpPr (Non-Visual Properties for a Group Shape) 3170
20.5.2.22 nvPicPr (Non-Visual Properties for a Picture) 3171
20.5.2.23 nvSpPr (Non-Visual Properties for a Shape) 3171
20.5.2.24 oneCellAnchor (One Cell Anchor Shape Size) 3171
20.5.2.25 pic (Picture) 3171
20.5.2.26 pos (Position) 3173
20.5.2.27 row (Row) 3173
20.5.2.28 rowOff (Row Offset) 3173
20.5.2.29 sp (Shape) 3174
20.5.2.30 spPr (Shape Properties) 3175
20.5.2.31 style (Shape Style) 3175
20.5.2.32 to (Ending Anchor Point) 3175
20.5.2.33 twoCellAnchor (Two Cell Anchor Shape Size) 3176
20.5.2.34 txBody (Shape Text Body) 3177
20.5.2.35 wsDr (Worksheet Drawing) 3177
20.5.2.36 xfrm (2D Transform for Graphic Frames) 3177

20.5.3 Simple Types 3178

20.5.3.1 ST_ColID (Column ID) 3178
20.5.3.2 ST_EditAs (Resizing Behaviors) 3178
20.5.3.3 ST_RowID (Row ID) 3179

End of informative text.

20.5.2 Elements

The following elements define the contents of the Spreadsheet Drawing namespace:

20.5.2.1 absoluteAnchor (Absolute Anchor Shape Size)

This element is used as an anchor placeholder for a shape or group of shapes. It anchors the object in the same position relative to sheet position and its extents are in EMU units.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AbsoluteAnchor) is located in §A.4.5. *end note*]

20.5.2.2 blipFill (Picture Fill)

This element specifies the type of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<xdr:pic>
  ..
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <xdr:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </xdr:blipFill>
    ..
  </xdr:pic>
```

The above picture object is shown as an example of this fill type. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
  <xdr:pic>
    ..
    <xdr:blipFill>
      <a:blip r:embed="rId2"/>
      <a:stretch>
        <a:fillRect/>
      </a:stretch>
    </xdr:blipFill>
    <xdr:spPr>
      <a:gradFill>
        <a:gsLst>
          <a:gs pos="0">
            <a:schemeClr val="tx2">
              <a:shade val="50000"/>
            </a:schemeClr>
          </a:gs>
          <a:gs pos="39999">
            <a:schemeClr val="tx2">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
        </a:gs>
      </a:gsLst>
    </a:gradFill>
  </xdr:spPr>
  ..
</xdr:pic>
```

The above picture object is shown as an example of this double fill type. *end example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: http://purl.oclc.or	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
g/ooxml/drawingml/main	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

20.5.2.3 clientData (Client Data)

This element is used to set certain properties related to a drawing element on the client spreadsheet application.

Attributes	Description
fLocksWithSheet (Locks With Sheet Flag)	This attribute indicates whether to disable selection on drawing elements when the sheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fPrintsWithSheet (Prints With Sheet Flag)	This attribute indicates whether to print drawing elements when printing the sheet. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnchorClientData) is located in §A.4.5. *end note*]

20.5.2.4 cNvCxnSpPr (Non-Visual Connector Shape Drawing Properties)

This element specifies the non-visual properties for a connector shape. These are the set of properties on a shape which do not affect its display within a spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

20.5.2.5 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual properties for a single graphical object frame within a spreadsheet. These are the set of properties of a frame which do not affect its display within a spreadsheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

20.5.2.6 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual properties of a hierarchical grouping of shapes, graphical object frames, and child groups. These are the set of properties of a group which do not affect its display within a spreadsheet.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

20.5.2.7 cNvPicPr (Non-Visual Picture Drawing Properties)

This element describes the non-visual properties of a picture within a spreadsheet. These are the set of properties of a picture which do not affect its display within a spreadsheet.

[*Example*: Consider the following SpreadsheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <xdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </xdr:cNvPicPr>
    <xdr:nvPr/>
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example defines some non-visual picture drawing properties for the inserted picture. *end example*]

Attributes	Description
preferRelativeResize (Relative Resize Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.5.2.8 cNvPr (Non-Visual Drawing Properties)

This element specifies the set of non-visual properties for the parent element. These properties specify all the data about the parent which does not affect its display within the spreadsheet.

[*Example*: Consider the following SpreadSheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example defines some non-visual drawing properties for the inserted picture. *end example*]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text which can be used in place of the actual DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >` The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM: Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

20.5.2.9 cNvSpPr (Connection Non-Visual Shape Properties)

This element specifies the set of non-visual properties for a connection shape. These properties specify all data about the connection shape which do not affect its display within a spreadsheet.

[*Example*: Consider the shape that has a shape lock applied to it.

```
<xdr:sp>
  <xdr:nvSpPr>
    <xdr:cNvPr id="2" name="Rectangle 1"/>
    <xdr:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </xdr:cNvSpPr>
   </xdr:nvSpPr>
   ..
</xdr:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box. [*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

20.5.2.10 col (Column))

This element specifies the column that is used within the from and to elements to specify anchoring information for a shape within a spreadsheet

The possible values for this element are defined by the ST_ColID simple type (§20.5.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_ColID) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

20.5.2.11 colOff (Column Offset)

This element is used to specify the column offset within a cell. The units for which this attribute is specified in reside within the simple type definition referenced below.

The possible values for this element are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.5.2.12 contentPart (Content Part)

This element specifies a reference to XML content in a format not defined by ISO/IEC 29500. [*Note*: This part allows the native use of other commonly used interchange formats, such as:

- MathML (http://www.w3.org/TR/MathML2/)
- SMIL (http://www.w3.org/TR/REC-smil/)
- SVG (http://www.w3.org/TR/SVG11/)

end note]

The relationship type of the explicit relationship specified by this element shall be http://purl.oclc.org/ooxml/officeDocument/relationships/customXml and have a TargetMode attribute value of `Internal`. If an application cannot process content of the content type specified by the targeted part, then it should continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

[*Example*: Consider a SpreadsheetML document which includes the following SVG markup in a part named svg1.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<!--======================================================================-->
<!--=  Copyright 2000 World Wide Web Consortium, (Massachusetts          =-->
<!--=  Institute of Technology, Institut National de Recherche en        =-->
<!--=  Informatique et en Automatique, Keio University). All Rights      =-->
<!--=  Reserved. See http://www.w3.org/Consortium/Legal/.                =-->
<!--======================================================================-->
<!-- ====================================================================-->
<!--                                                                     -->
<!-- color-datatypes-BE-01.svg                                           -->
<!-- renamed for 1.1 suite to color-prop-02-f.svg                        -->
<!--                                                                     -->
<!-- Author : Chris Lilley,  12-Aug-2000                                 -->
<!--          1.1 revision by Rick Graham                                -->
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<!--          Revised for SVGT/B: Benoit Bezaire Jul/02/2002             -->
<!--          More revision  CL  -->
<!--=======================================================================-->
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" id="svg-root" width="100%"
height="100%" viewBox="0 0 480 360">
  <title id="test-title">color-prop-02-f.svg</title>
  <desc id="test-desc">Test that viewer has the basic capability to render
X11colors, using any of the equivalent forms.</desc>
  <!--=======================================================================-->
  <!--Content of Test Case follows…                ==============-->
  <!--=======================================================================-->
  <g id="test-body-content">
    <!-- groups of five colors  -->
    <g>
      <circle cx="75" cy="40" r="20" fill="crimson"/>
      <circle cx="115" cy="40" r="20" fill="#DC143C"/>
      <circle cx="75" cy="80" r="20" fill="rgb(220,20,60)"/>
      <circle cx="115" cy="80" r="20"
fill="rgb(86.27450980392156862745098039215%,7.843137254901960784313725490196%,
23.529411764705882352941176470588%)"/>
    </g>
    <g>
      <circle cx="200" cy="40" r="20" fill="palegreen"/>
      <circle cx="240" cy="40" r="20" fill="#98FB98"/>
      <circle cx="200" cy="80" r="20" fill="rgb(152, 251, 152)"/>
      <circle cx="240" cy="80" r="20"
fill="rgb(59.607843137254901960784313725490%,98.431372549019607843137254901961%,5
9.607843137254901960784313725490%)"/>
    </g>
    <g>
      <circle cx="325" cy="40" r="20" fill="royalblue"/>
      <circle cx="365" cy="40" r="20" fill="#4169E1"/>
      <circle cx="325" cy="80" r="20" fill="rgb(65, 105, 225)"/>
      <circle cx="365" cy="80" r="20"
fill="rgb(25.490196078431372549019607843137%,41.176470588235294117647058823529%,
88.235294117647058823529411764706%)"/>
    </g>
    <g>
      <circle cx="75" cy="135" r="20" fill="firebrick"/>
      <circle cx="115" cy="135" r="20" fill="#B22222"/>
      <circle cx="75" cy="175" r="20" fill="rgb(178,34,34)"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <circle cx="115" cy="175" r="20"
fill="rgb(69.80392156862745098039215686274509%,13.333333333333333333333333333333%,
13.333333333333333333333333333333%)"/>
    </g>
    <g>
      <circle cx="200" cy="135" r="20" fill="seagreen"/>
      <circle cx="240" cy="135" r="20" fill="#2E8B57"/>
      <circle cx="200" cy="175" r="20" fill="rgb(46, 139, 87)"/>
      <circle cx="240" cy="175" r="20"
fill="rgb(18.039215686274509803921568627451%,54.509803921568627450980392156863%,
34.117647058823529411764705882353%)"/>
    </g>
    <g>
      <circle cx="325" cy="135" r="20" fill="mediumblue"/>
      <circle cx="365" cy="135" r="20" fill="#0000CD"/>
      <circle cx="325" cy="175" r="20" fill="rgb(0, 0, 205)"/>
      <circle cx="365" cy="175" r="20"
fill="rgb(0%,0%,80.392156862745098039215686274509%)"/>
    </g>
    <g>
      <circle cx="75" cy="230" r="20" fill="indianred"/>
      <circle cx="115" cy="230" r="20" fill="#CD5C5C"/>
      <circle cx="75" cy="270" r="20" fill="rgb(205, 92, 92)"/>
      <circle cx="115" cy="270" r="20"
fill="rgb(80.392156862745098039215686274509%,36.078431372549019607843137254902%,3
6.078431372549019607843137254902%)"/>
    </g>
    <g>
      <circle cx="200" cy="230" r="20" fill="lawngreen"/>
      <circle cx="240" cy="230" r="20" fill="#7CFC00"/>
      <circle cx="200" cy="270" r="20" fill="rgb(124, 252, 0)"/>
      <circle cx="240" cy="270" r="20"
fill="rgb(48.627450980392156862745098039216%,98.823529411764705882352941176471%,
0%)"/>
    </g>
    <g>
      <circle cx="325" cy="230" r="20" fill="mediumturquoise"/>
      <circle cx="365" cy="230" r="20" fill="#48D1CC"/>
      <circle cx="325" cy="270" r="20" fill="rgb(72, 209, 204)"/>
      <circle cx="365" cy="270" r="20"
fill="rgb(28.235294117647058823529411764706%,81.960784313725490196078431372549%,
80%)"/>
    </g>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
  </g>
  <text id="revision" x="10" y="340" font-size="40" stroke="none"
fill="black">$Revision: 1.6 $</text>
  <rect id="test-frame" x="1" y="1" width="478" height="358" fill="none"
stroke="#000000"/>
</svg>
```

The SpreadsheetML Drawing part would reference this content as follows:

```
<wsDr>
  <twoCellAnchor>
    <from>
      <col>3</col>
      <colOff>152400</colOff>
      <row>5</row>
      <rowOff>123825</rowOff>
    </from>
    <to>
      <col>8</col>
      <colOff>266700</colOff>
      <row>22</row>
      <rowOff>38100</rowOff>
    </to>
  </twoCellAnchor>
  <contentPart r:id="svg1"/>
</wsDr>
```

The contentPart element specifies that the SVG markup targeted by the relationship with an ID of svg1 is part of the SpreadsheetML document. Examining the contents of the corresponding relationship part item, we can see the targets for that relationship:

```
<Relationships … >
  …
  <Relationship Id="svg1" TargetMode="Internal"
Type="http://purl.oclc.org/ooxml/officeDocument/relationships/customXml"
Target="svg1.xml" />
  …
</Relationships>
```

The corresponding relationship part item shows that the file to be imported is named svg1.xml. *end example*]

Attributes	Description
id (Relationship to Part)	Specifies the relationship ID to a content part. [*Example*: Consider an XML element which has the following id attribute:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	<… r:id="rId1" /> The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. *end example*] The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rel) is located in §A.4.5. *end note*]

20.5.2.13 cxnSp (Connection Shape)

This element specifies the properties for a connection shape drawing element. A connection shape is a line, etc. that connects two other shapes in this drawing.

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Reference to Custom Function)	This element specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: <… macro="DoWork()" /> *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connector) is located in §A.4.5. *end note*]

20.5.2.14 ext (Shape Extent)

This element describes the length and width properties for how far a drawing element should extend for.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

20.5.2.15 from (Starting Anchor Point)

This element specifies the first anchor point for the drawing element. This is used to anchor the top and left sides of the shape within the spreadsheet. That is when the cell that is specified in the from element is adjusted, the shape is also adjusted.

[*Example*: Consider the following SpreadsheetDrawingML

```
<xdr:twoCellAnchor>
  <xdr:from>
    <xdr:col>3</xdr:col>
    <xdr:colOff>447675</xdr:colOff>
    <xdr:row>8</xdr:row>
    <xdr:rowOff>28575</xdr:rowOff>
  </xdr:from>
  <xdr:to>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <xdr:col>5</xdr:col>
      <xdr:colOff>466725</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>9525</xdr:rowOff>
    </xdr:to>
    <xdr:sp macro="" textlink="">
    …
    </xdr:sp>
    <xdr:clientData/>
  </xdr:twoCellAnchor>
```

The above example shows the first anchor point being specified via the from element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.4.5. *end note*]

20.5.2.16 graphicFrame (Graphic Frame)

This element describes a single graphical object frame for a spreadsheet which contains a graphical object.

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Reference To Custom Function)	This element specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrame) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.5.2.17 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<xdr:grpSp>
  <xdr:nvGrpSpPr>
    <xdr:cNvPr id="10" name="Group 9"/>
    <xdr:cNvGrpSpPr/>
    <xdr:nvPr/>
  </xdr:nvGrpSpPr>
  <xdr:grpSpPr>
    <a:xfrm>
      <a:off x="838200" y="990600"/>
      <a:ext cx="2426208" cy="978408"/>
      <a:chOff x="838200" y="990600"/>
      <a:chExt cx="2426208" cy="978408"/>
   </a:xfrm>
  </xdr:grpSpPr>
  <xdr:sp>
  ..
  </xdr:sp>
  <xdr:sp>
  ..
  </xdr:sp>
  <xdr:sp>
  ..
  </xdr:sp>
</xdr:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

20.5.2.18 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

20.5.2.19 nvCxnSpPr (Non-Visual Properties for a Connection Shape)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorNonVisual) is located in §A.4.5. *end note*]

20.5.2.20 nvGraphicFramePr (Non-Visual Properties for a Graphic Frame)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicalObjectFrameNonVisual) is located in §A.4.5. *end note*]

20.5.2.21 nvGrpSpPr (Non-Visual Properties for a Group Shape)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeNonVisual) is located in §A.4.5. *end note*]

20.5.2.22 nvPicPr (Non-Visual Properties for a Picture)

This element specifies all non-visual properties for a picture. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following SpreadsheetDrawingML.

```
<xdr:pic>
  ..
  <xdr:nvPicPr>
    ..
  </xdr:nvPicPr>
  ..
</xdr:pic>
```

The above example shows the defining of non-visual picture properties. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.4.5. *end note*]

20.5.2.23 nvSpPr (Non-Visual Properties for a Shape)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeNonVisual) is located in §A.4.5. *end note*]

20.5.2.24 oneCellAnchor (One Cell Anchor Shape Size)

This element specifies a one cell anchor placeholder for a group, a shape, or a drawing element. It moves with the cell and its extents is in EMU units.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OneCellAnchor) is located in §A.4.5. *end note*]

20.5.2.25 pic (Picture)

This element specifies the existence of a picture object within the spreadsheet.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following SpreadsheetDrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<xdr:pic>
  <xdr:nvPicPr>
    <xdr:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <xdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </xdr:cNvPicPr>
    <xdr:nvPr/>
  </xdr:nvPicPr>
  <xdr:blipFill>
  …
  </xdr:blipFill>
  <xdr:spPr>
  …
  </xdr:spPr>
</xdr:pic>
```

end example]

Attributes	Description
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Reference To Custom Function)	This element specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.5.2.26 pos (Position)

This element describes the position of a drawing element within a spreadsheet.

Attributes	Description
x (X-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The x attribute defines an x-coordinate of `0`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).
y (Y-Axis Coordinate) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a coordinate on the x-axis. The origin point for this coordinate shall be specified by the parent XML element. [*Example*: Consider the following point on a basic wrapping polygon for a DrawingML object: `<… x="0" y="100" />` The y attribute defines a y-coordinate of `100`. *end example*] The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Point2D) is located in §A.4.1. *end note*]

20.5.2.27 row (Row)

This element specifies the row that is used within the from and to elements to specify anchoring information for a shape within a spreadsheet.

The possible values for this element are defined by the ST_RowID simple type (§20.5.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_RowID) is located in §A.4.5. *end note*]

20.5.2.28 rowOff (Row Offset)

This element is used to specify the row offset within a cell. The units for which this attribute is specified in reside within the simple type definition referenced below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The possible values for this element are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Coordinate) is located in §A.4.1. *end note*]

20.5.2.29 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the SpreadsheetDrawingML framework. In addition to a geometry each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

Attributes	Description
fLocksText (Lock Text Flag)	This attribute indicates whether to allow text editing within this drawing object when the parent worksheet is protected. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fPublished (Publish to Server Flag)	This attribute indicates whether the shape shall be published with the worksheet when sent to the server. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
macro (Reference to Custom Function)	This element specifies the custom function associated with the object. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `< … macro="DoWork()" />` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
textlink (Text Link)	This attribute specifies a formula linking to spreadsheet cell data. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

20.5.2.30 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a special shape such as a connector shape or picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe additional object-specific properties within a document. This allows for these shapes to have both the properties of a shape as well as specific properties that are unique to only them.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

20.5.2.31 style (Shape Style)

The element specifies the style that is applied to a shape and the corresponding references for each of the style components such as lines and fills.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

20.5.2.32 to (Ending Anchor Point)

This element specifies the second anchor point for the drawing element. This is used to anchor the bottom and right sides of the shape within the spreadsheet. That is when the cell that is specified in the to element is adjusted, the shape is also adjusted.

[*Example*: Consider the following SpreadsheetDrawingML

```
<xdr:twoCellAnchor>
  <xdr:from>
    <xdr:col>3</xdr:col>
    <xdr:colOff>447675</xdr:colOff>
    <xdr:row>8</xdr:row>
    <xdr:rowOff>28575</xdr:rowOff>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xdr:from>
    <xdr:to>
      <xdr:col>5</xdr:col>
      <xdr:colOff>466725</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>9525</xdr:rowOff>
    </xdr:to>
    <xdr:sp macro="" textlink="">
    …
    </xdr:sp>
    <xdr:clientData/>
  </xdr:twoCellAnchor>
```

The above example shows the second anchor point being specified via the to element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.4.5. *end note*]

20.5.2.33 twoCellAnchor (Two Cell Anchor Shape Size)

This element specifies a two cell anchor placeholder for a group, a shape, or a drawing element. It moves with cells and its extents are in EMU units.

Attributes	Description

editAs (Positioning and Resizing Behaviors)	Specifies how the DrawingML contents shall be moved and/or resized when the rows and columns between its start and ending anchor (the from and to child elements) are resized, or have additional rows/columns inserted within them, or additional row/columns are added before them. The behaviors are discussed in the simple type referenced below.
	
	If this attribute is omitted, then its default value shall be assumed to be `twoCell`.
	
	[*Example*: Consider a drawing defined as follows:
	
	`<ws:twoCellAnchor editAs="absolute">`
	`…`
	`</ws:twoCellAnchor>`
	
	The editAs attribute has a value of `absolute`, which specifies that the sizing of this object must not change, instead the anchor locations should be moved as needed to maintain the same size and position. *end example*]
	
	The possible values for this attribute are defined by the ST_EditAs simple type (§20.5.3.2).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwoCellAnchor) is located in §A.4.5. *end note*]

20.5.2.34 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

20.5.2.35 wsDr (Worksheet Drawing)

This element specifies all drawing objects within the worksheet. It acts much like the spTree element within the DrawingML framework. Allowing for the specification of all shapes for a given part of a document, in this case a single Worksheet.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.4.5. *end note*]

20.5.2.36 xfrm (2D Transform for Graphic Frames)

This element specifies a two dimensional transform for a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	 end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

20.5.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – SpreadsheetML Drawing.

20.5.3.1 ST_ColID (Column ID)

This simple type specifies a column identification. The numerical value used for the column id should be non-negative and never exceed the number of total columns within the spreadsheet document.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ColID) is located in §A.4.5. *end note*]

20.5.3.2 ST_EditAs (Resizing Behaviors)

This simple type specifies all possible settings for how DrawingML contents shall be resized when the rows and columns between its start and ending anchor (the from and to child elements) are resized, or have additional rows/columns inserted within them.

[*Example*: Consider a drawing defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<ws:twoCellAnchor editAs="absolute">
  …
</ws:twoCellAnchor>
```

The editAs attribute has a value of absolute, which specifies that the sizing of this object must not change, instead the anchor locations should be moved as needed to maintain the same size. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
absolute (Do Not Move or Resize With Underlying Rows/Columns)	Specifies that the current start and end positions shall be maintained with respect to the distances from the absolute start point of the worksheet. If additional rows/columns are added before the drawing, the drawing shall move its anchors as needed to maintain this same absolute position.
oneCell (Move With Cells but Do Not Resize)	Specifies that the current drawing shall move with its row and column (i.e. the object is anchored to the actual from row and column), but that the size shall remain absolute. If additional rows/columns are added between the from and to locations of the drawing, the drawing shall move its to anchors as needed to maintain this same absolute size.
twoCell (Move and Resize With Anchor Cells)	Specifies that the current drawing shall move and resize to maintain its row and column anchors (i.e. the object is anchored to the actual from and to row and column).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_EditAs) is located in §A.4.5. *end note*]

20.5.3.3 ST_RowID (Row ID)

This simple type specifies a row identification. The numerical value used for the row id should be non-negative and never exceed the number of total rows within the spreadsheet document.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RowID) is located in §A.4.5. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21. DrawingML - Components Reference Material

The subordinate subclauses specify the semantics for the XML markup comprising DrawingML content, which can be used within the contents of WordprocessingML, SpreadsheetML, or PresentationML documents.

This portion of DrawingML describes components built on top of the previously defined framework to create charts, diagrams, etc..

21.1 DrawingML - Main

The following parts of the DrawingML Main namespace define additional base constructs for all kinds of DrawingML objects (e.g., paragraphs, text, tables, etc.).

21.1.1 Table of Contents

This subclause is informative.

21.1.2 Paragraphs and Rich Formatting 3183
21.1.2.1 Body Formatting 3183
21.1.2.1.1 bodyPr (Body Properties) 3184
21.1.2.1.2 noAutofit (No AutoFit) 3192
21.1.2.1.3 normAutofit (Normal AutoFit) 3193
21.1.2.1.4 spAutoFit (Shape AutoFit) 3194
21.1.2.2 Paragraph Formatting 3195
21.1.2.2.1 br (Text Line Break) 3196
21.1.2.2.2 defPPr (Default Paragraph Style) 3197
21.1.2.2.3 endParaRPr (End Paragraph Run Properties) 3204
21.1.2.2.4 fld (Text Field) 3209
21.1.2.2.5 lnSpc (Line Spacing) 3211
21.1.2.2.6 p (Text Paragraphs) 3212
21.1.2.2.7 pPr (Text Paragraph Properties) 3212
21.1.2.2.8 rtl (Right to Left Run) 3220
21.1.2.2.9 spcAft (Space After) 3222
21.1.2.2.10 spcBef (Space Before) 3223
21.1.2.2.11 spcPct (Spacing Percent) 3224
21.1.2.2.12 spcPts (Spacing Points) 3224
21.1.2.2.13 tab (Tab Stop) 3225
21.1.2.2.14 tabLst (Tab List) 3226
21.1.2.3 Run Formatting 3227
21.1.2.3.1 cs (Complex Script Font) 3228
21.1.2.3.2 defRPr (Default Text Run Properties) 3230
21.1.2.3.3 ea (East Asian Font) 3235
21.1.2.3.4 highlight (Highlight Color) 3237

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.1.2.3.5 hlinkClick (Click Hyperlink) 3238
21.1.2.3.6 hlinkMouseOver (Mouse-Over Hyperlink) 3240
21.1.2.3.7 latin (Latin Font) 3241
21.1.2.3.8 r (Text Run) 3243
21.1.2.3.9 rPr (Text Run Properties) 3244
21.1.2.3.10 sym (Symbol Font) 3249
21.1.2.3.11 t (Text String) 3251
21.1.2.3.12 uFill (Underline Fill) 3251
21.1.2.3.13 uFillTx (Underline Fill Properties Follow Text) 3252
21.1.2.3.14 uLn (Underline Stroke) 3253
21.1.2.3.15 uLnTx (Underline Follows Text) 3254
21.1.2.4 Bullets and Numbering 3254
21.1.2.4.1 buAutoNum (Auto-Numbered Bullet) 3254
21.1.2.4.2 buBlip (Picture Bullet) 3256
21.1.2.4.3 buChar (Character Bullet) 3257
21.1.2.4.4 buClr (Color Specified) 3259
21.1.2.4.5 buClrTx (Follow Text) 3259
21.1.2.4.6 buFont (Specified) 3260
21.1.2.4.7 buFontTx (Follow text) 3262
21.1.2.4.8 buNone (No Bullet) 3263
21.1.2.4.9 buSzPct (Bullet Size Percentage) 3263
21.1.2.4.10 buSzPts (Bullet Size Points) 3264
21.1.2.4.11 buSzTx (Bullet Size Follows Text) 3265
21.1.2.4.12 lstStyle (Text List Styles) 3266
21.1.2.4.13 lvl1pPr (List Level 1 Text Style) 3266
21.1.2.4.14 lvl2pPr (List Level 2 Text Style) 3274
21.1.2.4.15 lvl3pPr (List Level 3 Text Style) 3282
21.1.2.4.16 lvl4pPr (List Level 4 Text Style) 3290
21.1.2.4.17 lvl5pPr (List Level 5 Text Style) 3298
21.1.2.4.18 lvl6pPr (List Level 6 Text Style) 3306
21.1.2.4.19 lvl7pPr (List Level 7 Text Style) 3314
21.1.2.4.20 lvl8pPr (List Level 8 Text Style) 3322
21.1.2.4.21 lvl9pPr (List Level 9 Text Style) 3330
21.1.2.5 Font Substitution 3338

21.1.3 Tables 3338

21.1.3.1 cell3D (Cell 3-D) 3338
21.1.3.2 gridCol (Table Grid Column) 3339
21.1.3.3 header (Header Cell Reference) 3339
21.1.3.4 headers (Header Cells Associated With Table Cell) 3341
21.1.3.5 lnB (Bottom Border Line Properties) 3342
21.1.3.6 lnBlToTr (Bottom-Left to Top-Right Border Line Properties) 3343
21.1.3.7 lnL (Left Border Line Properties) 3344
21.1.3.8 lnR (Right Border Line Properties) 3345
21.1.3.9 lnT (Top Border Line Properties) 3346
21.1.3.10 lnTlToBr (Top-Left to Bottom-Right Border Line Properties) 3347
21.1.3.11 tableStyle (Table Style) 3348
21.1.3.12 tableStyleId (Table Style ID) 3350

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.1.3.13 tbl (Table) 3351
21.1.3.14 tblGrid (Table Grid) 3351
21.1.3.15 tblPr (Table Properties) 3351
21.1.3.16 tc (Table Cell) 3354
21.1.3.17 tcPr (Table Cell Properties) 3357
21.1.3.18 tr (Table Row) 3360

End of informative text.

21.1.2 Paragraphs and Rich Formatting

The Paragraphs and Rich Formatting portion of the DrawingML framework stores text and related formatting information for a text body contained within a shape. Formatting for text within a shape can be broken down into three levels of precision, namely body, paragraph, and run formatting properties.

21.1.2.1 Body Formatting

Being the highest level of formatting available within a shape, the body properties allow for the manipulation of the text area as a whole. This means that all paragraphs and runs of text for the shape in question would be encompassed within here and, therefore, follow the text body style defined here.

[*Example*: Consider a shape that has three paragraphs within it, each with a different sized text. If this shape is resized to be smaller, then the text no longer fits the same way within the shape. Thus, we see that to maintain visual quality the size must be changed.

Rather than try to change each of the paragraphs to a smaller font size to fit in the shape, just use a body-level format property such as the normAutofit. This applies to all text within the shape and scale the text by a certain percentage in order to fit within the shape.

```
<p:txBody>
  <a:bodyPr>
    <a:normAutofit fontScale="20.000%" lnSpcReduction="20.000%"/>
  </a:bodyPr>
  …
  <a:p>
  …
  </a:p>
  <a:p>
  …
  </a:p>
  <a:p>
  …
  </a:p>
</p:txBody>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.1.2.1.1 bodyPr (Body Properties)

This element defines the body properties for the text body within a shape.

[*Example*: Consider a shape with a text body that has some formatting properties associated with it. For the formatting of text body properties, the bodyPr element should be used as follows:

```
<p:sp>
  …
  <p:txBody>
    <a:bodyPr>
      (text body properties)
    </a:bodyPr>
  …
  </p:txBody>
</p:sp>
```

end example]

Attributes	Description

anchor (Anchor)	Specifies the anchoring position of the txBody within the shape. If this attribute is omitted, then a value of t, or top is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr anchor="ctr" … />`
	`  …`
	`</p:txBody>`
	
	Here the text is vertically aligned in the center of the shape within which it is contained. *end example*]
	
	The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.60).

anchorCtr (Anchor Center)	Specifies the centering of the text box. The way it works fundamentally is to determine the smallest possible "bounds box" for the text and then to center that "bounds box" accordingly. This is different than paragraph alignment, which aligns the text within the "bounds box" for the text. This flag is compatible with all of the different kinds of anchoring. If this attribute is omitted, then a value of `0` or `false` is implied.
	
	[*Example*: The text within this shape has been both vertically centered with the anchor attribute and horizontally centered with the anchorCtr attribute.
	
	`<p:txBody>`
	`  <a:bodyPr anchor="ctr" anchorCtr="1" … />`
	`  …`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
bIns (Bottom Inset)	Specifies the bottom inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, a value of 45720 or 0.05 inches is implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr lIns="91440" tIns="91440" rIns="91440" bIns="91440" … /> … </p:txBody> The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
compatLnSpc (Compatible Line Spacing)	Specifies that the line spacing for this text body is decided in a simplistic manner using the font scene. If this attribute is omitted, a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
forceAA (Force Anti-Alias)	Forces the text to be rendered anti-aliased regardless of the font size. Certain fonts can appear grainy around their edges unless they are anti-aliased. Therefore this attribute allows for the specifying of which bodies of text should always be anti-aliased and which ones should not. If this attribute is omitted, then a value of `0` or `false`is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fromWordArt (From WordArt)	Specifies that text within this textbox is converted text from a WordArt object. This is more of a backwards compatibility attribute that is useful to the application from a tracking perspective. WordArt was the former way to apply text effects and therefore this attribute is useful in document conversion scenarios. If this attribute is omitted, then a value of `0` or `false`is implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr wrap="none" fromWordArt="1" … /> … </p:txBody>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] Because of the presence of the fromWordArt attribute the text within this shape can be mapped back to the corresponding WordArt during document conversion. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

horzOverflow (Text Horizontal Overflow)	Determines whether the text can flow out of the bounding box horizontally. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied.
	
	[*Example*: Consider the case where we have multiply paragraphs within a shape and the second is greater in length and causes text to flow outside the shape. By applying the clip value of the horzOverflow attribute as a body property this overflowing text now is cut off instead of extending beyond the bounds of the shape.
	
	`<p:txBody>`
	`  <a:bodyPr horzOverflow="clip" … />`
	`  …`
	`  <a:p>`
	`  …`
	`  (Some text)`
	`  …`
	`  </a:p>`
	`  <a:p>`
	`  …`
	`  (Some more text)`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.69).

lIns (Left Inset)	Specifies the left inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr lIns="91440" tIns="91440" rIns="91440"`
	`  bIns="91440" … />`
	`  …`
	`</p:txBody>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

numCol (Number of Columns)	Specifies the number of columns of text in the bounding rectangle. When applied to a text run this property takes the width of the bounding box for the text and divides it by the number of columns specified. These columns are then treated as overflow containers in that when the previous column has been filled with text the next column acts as the repository for additional text. When all columns have been filled and text still remains then the overflow properties set for this text body are used and the text is reflowed to make room for additional text. If this attribute is omitted, then a value of 1 is implied.
	
	[*Example*: Consider the case where a text area would need to be split up into four separate columns. Then simply specifying one paragraph with one run of text is enough to describe four columns of text here.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="4" … />`
	`  <a:p>`
	`  <a:r>`
	`  …`
	`  (Some text)`
	`  …`
	`  </a:r>`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextColumnCount simple type (§20.1.10.65).

rIns (Right Inset)	Specifies the right inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 91440 or 0.1 inches is implied.
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr lIns="91440" tIns="91440"`
	`  rIns="91440" bIns="91440" … />`
	`  …`
	`</p:txBody>`
	
	The text box having the above body properties has inset margins of 91440 or 0.1 inches

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

rot (Rotation)	Specifies the rotation that is being applied to the text within the bounding box. If it not specified, the rotation of the accompanying shape is used. If it is specified, then this is applied independently from the shape. That is the shape can have a rotation applied in addition to the text itself having a rotation applied to it. If this attribute is omitted, then a value of 0, is implied.
	
	[*Example*: Consider the case where a shape has a rotation of 5400000, or 90 degrees clockwise applied to it. In addition to this, the text body itself has a rotation of -5400000, or 90 degrees counter-clockwise applied to it. Then the resulting shape would appear to be rotated but the text within it would appear as though it had not been rotated at all. The DrawingML specifying this would look like the following:
	
	`<p:sp>`
	`  <p:spPr>`
	`  <a:xfrm rot="5400000">`
	`  …`
	`  </a:xfrm>`
	`  </p:spPr>`
	`  …`
	`  <p:txBody>`
	`  <a:bodyPr rot="-5400000" … />`
	`  …`
	`  (Some text)`
	`  …`
	`  </p:txBody>`
	`</p:sp>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

rtlCol (Columns Right-To-Left)	Specifies whether columns are used in a right-to-left or left-to-right order. The usage of this attribute only sets the column order that is used to determine which column overflow text should go to next. If this attribute is omitted, then a value of `0` or `false`is implied in which case text starts in the leftmost column and flow to the right. [*Note*: This attribute in no way determines the direction of text but merely the direction in which multiple columns are used. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
spcCol (Space	Specifies the space between text columns in the text area. This should only apply when

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Between Columns)	there is more than 1 column present. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_PositiveCoordinate32 simple type (§20.1.10.43).

spcFirstLastPara (Paragraph Spacing)	Specifies whether the before and after paragraph spacing defined by the user is to be respected. While the spacing between paragraphs is helpful, it is additionally useful to be able to set a flag as to whether this spacing is to be followed at the edges of the text body, in other words the first and last paragraphs in the text body. More precisely since this is a text body level property it should only effect the before paragraph spacing of the first paragraph and the after paragraph spacing of the last paragraph for a given text body. If this attribute is omitted, then a value of 0, or false is implied.
	
	[*Example*: Consider the case where spacing has been defined between multiple paragraphs within a text body using the spcBef and spcAft paragraph spacing attributes. For this text body however the user would like to not have this followed for the edge paragraphs and thus we have the following DrawingML.
	
	`<p:txBody>`
	`<a:bodyPr spcFirstLastPara="0" … />`
	`…`
	`<a:p>`
	`<a:pPr>`
	`<a:spcBef>`
	`<a:spcPts val="1800"/>`
	`</a:spcBef>`
	`<a:spcAft>`
	`<a:spcPts val="600"/>`
	`</a:spcAft>`
	`</a:pPr>`
	`…`
	`(Some text)`
	`…`
	`</a:p>`
	`<a:p>`
	`<a:pPr>`
	`<a:spcBef>`
	`<a:spcPts val="1800"/>`
	`</a:spcBef>`
	`<a:spcAft>`
	`<a:spcPts val="600"/>`
	`</a:spcAft>`
	`</a:pPr>`
	`…`
	`(Some text)`
	`…`
	`</a:p>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
tIns (Top Inset)	Specifies the top inset of the bounding rectangle. Insets are used just as internal margins for text boxes within shapes. If this attribute is omitted, then a value of 45720 or 0.05 inches is implied. [*Example*: Consider the following DrawingML: <p:txBody> <a:bodyPr lIns="91440" tIns="91440" rIns="91440" bIns="91440" … /> … </p:txBody> The text box having the above body properties has inset margins of 91440 or 0.1 inches on all four sides. *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
upright (Text Upright)	Specifies whether text should remain upright, regardless of the transform applied to it and the accompanying shape transform. If this attribute is omitted, then a value of 0, or false is implied. [*Example*: Consider text that has been rotated within the text body but has the upright flag set. <p:txBody> <a:bodyPr upright="1" rot="5400000" …/> … (Some text) … </p:txBody> This text appears as though no transform has been applied to it. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
vert (Vertical Text)	Determines if the text within the given text body should be displayed vertically. If this attribute is omitted, then a value of horz, or no vertical text is implied. [*Example*: Consider the case where the user needs to display text that appears vertical and has a right to left flow with respect to its columns.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<p:txBody> <a:bodyPr vert="wordArtVertRtl" … /> … <a:p> … <a:t>This is</a:t> … </a:p> <a:p> … <a:t>some text.</a:t> … </a:p> </p:txBody> In the above sample DrawingML there are two paragraphs denoting a separation between the text otherwise which are known as either a line or paragraph break. Because wordArtVertRtl is used here this text is not only displayed in a stacked manner flowing from top to bottom but also have the first paragraph be displayed to the right of the second. This is because it is both vertical text and right to left. *end example*] The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.83).
vertOverflow (Text Vertical Overflow)	Determines whether the text can flow out of the bounding box vertically. This is used to determine what happens in the event that the text within a shape is too large for the bounding box it is contained within. If this attribute is omitted, then a value of overflow is implied. [*Example*: Consider the case where we have multiply paragraphs within a shape and the second causes text to flow outside the shape. By applying the clip value of the vertOverflow attribute as a body property this overflowing text is now cut off instead of extending beyond the bounds of the shape. <p:txBody> <a:bodyPr vertOverflow="clip" … /> … <a:p> … (Some text) … </a:p> <a:p> … (Some longer text) … </a:p>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextVertOverflowType simple type (§20.1.10.84).
wrap (Text Wrapping Type)	Specifies the wrapping options to be used for this text body. If this attribute is omitted, then a value of square is implied which wraps the text using the bounding text box. The possible values for this attribute are defined by the ST_TextWrappingType simple type (§20.1.10.85).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBodyProperties) is located in §A.4.1. *end note*]

21.1.2.1.2 noAutofit (No AutoFit)

This element specifies that text within the text body should not be auto-fit to the bounding box. Auto-fitting is when text within a text box is scaled in order to remain inside the text box. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider a text box where the user wishes to have the text extend outside the bounding box. The following DrawingML would describe this.

```
<p:txBody>
  <a:bodyPr wrap="none" rtlCol="0">
    <a:noAutofit/>
  </a:bodyPr>
  <a:p>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNoAutofit) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.1.2.1.3 normAutofit (Normal AutoFit)

This element specifies that text within the text body should be normally auto-fit to the bounding box. Auto-fitting is when text within a text box is scaled in order to remain inside the text box. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider the situation where a user is building a diagram and needs to have the text for each shape that they are using stay within the bounds of the shape. An easy way this might be done is by using normAutofit. The following DrawingML illustrates how this might be accomplished.

```
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:normAutofit fontScale="92.000%" lnSpcReduction="20.000%"/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 1</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:normAutofit fontScale="92.000%" lnSpcReduction="20.000%"/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 2</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

In the above example there are two shapes that have normAutofit turned on so that when the user types more text within the shape that the text actually resizes to accommodate the new data. For the application to know how and to what degree the text should be resized two attributes are set for the auto-fit resize logic. *end example*]

Attributes	Description
fontScale (Font	Specifies the percentage of the original font size to which each run in the text body is

Attributes	Description
Scale)	scaled. In order to auto-fit text within a bounding box it is sometimes necessary to decrease the font size by a certain percentage. Using this attribute the font within a text box can be scaled based on the value provided. A value of 100% scales the text to 100%, while a value of 1% scales the text to 1%. If this attribute is omitted, then a value of 100% is implied. The possible values for this attribute are defined by the ST_TextFontScalePercentOrPercentString simple type (§20.1.10.67).
lnSpcReduction (Line Space Reduction)	Specifies the percentage amount by which the line spacing of each paragraph in the text body is reduced. The reduction is applied by subtracting it from the original line spacing value. Using this attribute the vertical spacing between the lines of text can be scaled by a percent amount. A value of 100% reduces the line spacing by 100%, while a value of 1% reduces the line spacing by one percent. If this attribute is omitted, then a value of 0% is implied. [*Note*: This attribute applies only to paragraphs with percentage line spacing. *end note*] The possible values for this attribute are defined by the ST_TextSpacingPercentOrPercentString simple type (§20.1.10.77).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNormalAutofit) is located in §A.4.1. *end note*]

21.1.2.1.4 spAutoFit (Shape AutoFit)

This element specifies that a shape should be auto-fit to fully contain the text described within it. Auto-fitting is when text within a shape is scaled in order to contain all the text inside. If this element is omitted, then noAutofit or auto-fit off is implied.

[*Example*: Consider the situation where a user is building a diagram and needs to have the text for each shape that they are using stay within the bounds of the shape. An easy way this might be done is by using spAutofit. The following DrawingML illustrates how this might be accomplished.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:spAutoFit/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 1</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
<p:sp>
  <p:txBody>
    <a:bodyPr rtlCol="0" anchor="ctr">
      <a:spAutoFit/>
    </a:bodyPr>
    …
    <a:p>
      …
      <a:t>Diagram Object 2</a:t>
      …
    </a:p>
  </p:txBody>
</p:sp>
```

In the above example there are two shapes that have spAutoFit turned on so that when the user types more text within the shape that the shape actually resizes to accommodate the new data. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextShapeAutofit) is located in §A.4.1. *end note*]

21.1.2.2 Paragraph Formatting

This level of formatting allows for more granular control of text within a shape. Properties here apply to all text residing within the corresponding paragraph. This intermediate property level allows freedom to assign what would seem like lower level properties to a larger group of text. Along with this the paragraph property level also allows what would seem like larger group properties to a more granular set of text. This makes for a property level that is quite versatile in its ability to define formatting on text within a shape.

[*Example*: For instance consider the case where a paragraph of text would need to have bullets applied to it. At first one might think that this formatting must be done at the text run level as it can seem run specific. Much to the contrary this is a paragraph level property and is applied to multiple runs of text. As an example we have once again our three paragraphs with the second having bullets applied to it.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<a:p>
…
</a:p>
<a:p>
  <a:pPr>
    <a:buFont typeface="Wingdings"/>
    <a:buChar typeface="ü"/>
  <a:pPr>
  <a:r>
    <a:rPr lang="en-US" dirty="0" smtClean="0"/>
    <a:t>This Paragraph of Text Will Have a Bullet.</a:t>
  </a:r>
</a:p>
<a:p>
…
</a:p>
```

Here we see that the paragraph is formatted to have character bullets for each new line of text that is encountered. In particular this paragraph has the "ü" character applied which in the "Wingdings" font is the checkmark character. The other paragraphs are not effected by this paragraph's bullet formatting and should have their text remain unformatted.

end example]

21.1.2.2.1 br (Text Line Break)

This element specifies the existence of a vertical line break between two runs of text within a paragraph. In addition to specifying a vertical space between two runs of text, this element can also have run properties specified via the rPr child element. This sets the formatting of text for the line break so that if text is later inserted there that a new run can be generated with the correct formatting.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      …
      <a:t>Text Run 1.</a:t>
      …
    </a:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <a:br/>
      <a:r>
        …
        <a:t>Text Run 2.</a:t>
        …
      </a:r>
    </a:p>
  </p:txBody>
```

This paragraph has two runs of text laid out in a vertical fashion with a line break in between them. This line break acts much like a carriage return would within a normal run of text. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextLineBreak) is located in §A.4.1. *end note*]

21.1.2.2.2 defPPr (Default Paragraph Style)

This element specifies the paragraph properties that are to be applied when no other paragraph properties have been specified. If this attribute is omitted, then it is left to the application to decide the set of default paragraph properties that should be applied.

[*Example*: Consider the DrawingML shown below.

```
  <p:txBody>
    …
    <a:lstStyle>
      <a:defPPr>
        <a:buNone/>
      </a:defPPr>
    </a:lstStyle>
    <a:p>
      …
      <a:t>Sample Text</a:t>
      …
    </a:p>
  </p:txBody>
```

The above paragraph follows the properties described in defPPr if no overriding properties are specified within the pPr element. *end example*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

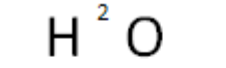
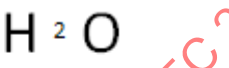
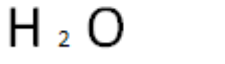
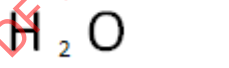
Attributes	Description
	Sample text Sample text Sample text

Attributes	Description
	<a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> … <a:p> <a:pPr eaLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	 
	fontAlgn="t"
	 
	fontAlgn="ctr"
	 
	fontAlgn="base"
	 
	fontAlgn="b"
	
	`<a:txtBody>`
	`  …`
	`  <a:pPr fontAlgn="b" …/>`
	`  …`
	`  <a:r>`
	`  <a:rPr …/>`
	`  <a:t>H </a:t>`
	`  </a:r>`
	`  <a:r>`
	`  <a:rPr sz="1200" …/>`
	`  <a:t>2</a:t>`
	`  </a:r>`
	`  <a:r>`
	`  <a:rPr …/>`
	`  <a:t>O</a:t>`
	`  </a:r>`
	`  …`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.
	
	Here is some text Sample text
	
	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`<a:bodyPr numCol="2" spcCol="914400"…/>`
	`<a:normAutofit/>`
	`</a:bodyPr>`
	`…`
	`<a:p>`
	`<a:pPr marL="0" indent="571500" algn="just">`
	`<a:buNone/>`
	`</a:pPr>`
	`…`
	`<a:t>Here is some…</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.70).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr
	agilisticexpialidocious
	
	Sample text Sample text Sample text
	supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values. [*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<p:txBody> … <a:p> <a:pPr lvl="1" …/> … <a:t>Sample text</a:t> … </a:p> </p:txBody> *end example*] [*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied. [*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1"/>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.2.3 endParaRPr (End Paragraph Run Properties)

This element specifies the text run properties that are to be used if another run is inserted after the last run specified. This effectively saves the run property state so that it can be applied when the user enters additional text. If this element is omitted, then the application can determine which default properties to apply. It is

©ISO/IEC 2016 – All rights reserved

recommended that this element be specified at the end of the list of text runs within the paragraph so that an orderly list is maintained.

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed.
	[*Example*: Consider the DrawingML shown below.
	`<a:p>`
	`…`
	`<a:rPr b="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	The above run of text is formatted as bold text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type (§20.1.10.64).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr i="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as italic text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font. The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.73).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  <a:r>`
	`  <a:rPr lang="en-US" dirty="0" smtId="1"/>`
	`  <a:t>CNTS</a:t>`
	`  </a:r>`
	`  <a:endParaRPr lang="en-US" dirty="0"/>`
	`  </a:p>`
	`</p:txBody>`
	
	The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.74).

strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr strike="sngStrike"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single strikethrough text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.79).

sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted with a 12 point text size. *end example*]
	
	The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.68).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr u="sng"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single underline text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.82).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.2.4 fld (Text Field)

This element specifies a text field which contains generated text that the application should update periodically. Each piece of text when it is generated is given a unique identification number that is used to refer to a specific field. At the time of creation the text field indicates the kind of text that should be used to update this field. This update type is used so that all applications that did not create this text field can still know what kind of text it should be updated with. Thus the new application can then attach an update type to the text field id for continual updating.

[*Example*: Consider a slide within a presentation that needs to have the slide number placed on the slide. The following DrawingML can be used to describe such a situation.

```
<p:txBody>
  <a:bodyPr/>
    <a:lstStyle/>
    <a:p>
      <a:fld id="{424CEEAC-8F67-4238-9622-1B74DC6E8318}" type="slidenum">
      <a:rPr lang="en-US" smtClean="0"/>
      <a:pPr/>
      <a:t>3</a:t>
    </a:fld>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <a:endParaRPr lang="en-US"/>
    </a:p>
  </p:txBody>
```

end example]

Attributes	Description
id (Field ID)	Specifies the unique to this document, host specified token that is used to identify the field. This token is generated when the text field is created and persists in the file as the same token until the text field is removed. Any application should check the document for conflicting tokens before assigning a new token to a text field. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
type (Field Type)	Specifies the type of text that should be used to update this text field. This is used to inform the rendering application what text it should use to update this text field. There are no specific syntax restrictions placed on this attribute. The generating application can use it to represent any text that should be updated before rendering the presentation. Reserved Values:

Value	Description
slidenum	presentation slide number
datetime	default date time format for the rendering application
datetime1	MM/DD/YYYY date time format [*Example:* 10/12/2007 *end example*]
datetime2	Day, Month DD, YYYY date time format [*Example:* Friday, October 12, 2007 *end example*]
datetime3	DD Month YYYY date time format [*Example:* 12 October 2007 *end example*]
datetime4	Month DD, YYYY date time format [*Example:* October 12, 2007 *end example*]
datetime5	DD-Mon-YY date time format [*Example:* 12-Oct-07 *end example*]
datetime6	Month YY date time format [*Example:* October 07 *end example*]
datetime7	Mon-YY date time format [*Example:* Oct-07 *end example*]
datetime8	MM/DD/YYYY hh:mm AM/PM date time format [*Example:* 10/12/2007 4:28 PM *end example*]
datetime9	MM/DD/YYYY hh:mm:ss AM/PM date time format [*Example:* 10/12/2007 4:28:34 PM *end example*]
datetime10	hh:mm date time format [*Example:* 16:28 *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td>datetime11</td><td>hh:mm:ss date time format [*Example:* 16:28:34 *end example*]</td></tr><tr><td>datetime12</td><td>hh:mm AM/PM date time format [*Example:* 4:28 PM *end example*]</td></tr><tr><td>datetime13</td><td>hh:mm:ss: AM/PM date time format [*Example:* 4:28:34 PM *end example*]</td></tr></table> The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextField) is located in §A.4.1. *end note*]

21.1.2.2.5 lnSpc (Line Spacing)

This element specifies the vertical line spacing that is to be used within a paragraph. This can be specified in two different ways, percentage spacing and font point spacing. If this element is omitted then the spacing between two lines of text should be determined by the point size of the largest piece of text within a line.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  <a:p>
    <a:pPr>
      <a:lnSpc>
        <a:spcPct val="200%"/>
      </a:lnSpc>
    </a:pPr>
    <a:r>
      <a:rPr lang="en-US" dirty="0" smtClean="0"/>
      <a:t>Some</a:t>
    </a:r>
    <a:br>
      <a:rPr lang="en-US" smtClean="0"/>
    </a:br>
    <a:r>
      <a:rPr lang="en-US" dirty="0" smtClean="0"/>
      <a:t>Text</a:t>
    </a:r>
  </a:p>
</p:txBody>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This paragraph has two lines of text that have percentage based vertical spacing. This kind of spacing should change based on the size of the text involved as its size is calculated as a percentage of this. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

21.1.2.2.6 p (Text Paragraphs)

This element specifies the presence of a paragraph of text within the containing text body. The paragraph is the highest level text separation mechanism within a text body. A paragraph can contain text paragraph properties associated with the paragraph. If no properties are listed then properties specified in the defPPr element are used.

[*Example*: Consider the case where the user would like to describe a text body that contains two paragraphs. The requirement for these paragraphs is that one be right aligned and the other left aligned. The following DrawingML would specify a text body such as this.

```
<p:txBody>
    …
    <a:p>
    <a:pPr algn="r">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr algn="l">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraph) is located in §A.4.1. *end note*]

21.1.2.2.7 pPr (Text Paragraph Properties)

This element contains all paragraph level text properties for the containing paragraph. These paragraph properties should override any and all conflicting properties that are associated with the paragraph in question.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:p>
  <a:pPr marL="0" algn="ctr">
    <a:buNone/>
  </a:pPr>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

The paragraph described above is formatting with a left margin of 0 and has all of text runs contained within it centered about the horizontal median of the bounding box for the text body. *end example*]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

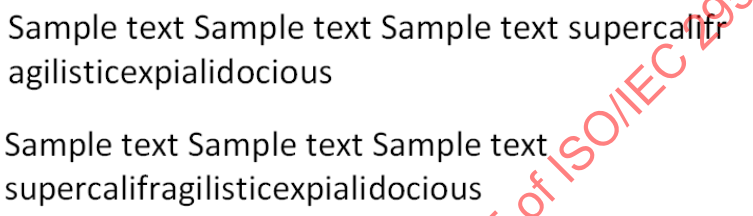
Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text Sample text Sample text

Attributes	Description
	<a:pPr marL="0" algn="just"> <a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

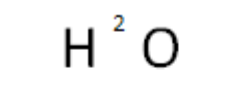
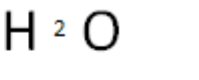
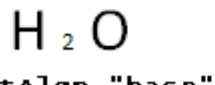
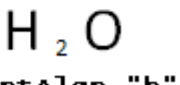
	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	
	fontAlgn="t"
	 
	fontAlgn="ctr"
	 
	fontAlgn="base"
	 
	fontAlgn="b"
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`
	`</a:r>`
	`…`
	`</p:txBody>`
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).

hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	attribute is omitted, then a value of -342900 is implied. Here is some text Sample text Sample text Sample

Attributes	Description

	be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr lvl="1" …/>`
	`  …`
	`  <a:t>Sample text</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Test تجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:t>تجربة_</a:t> </a:r> </a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.2.8 rtl (Right to Left Run)

This element specifies whether the contents of this run shall have right-to-left characteristics. Specifically, the following behaviors are applied when this element's val attribute is `true` (or an equivalent):

- Formatting – When the contents of this run are displayed, all characters shall be treated as complex script characters. This means that the values of the cs element (§21.1.2.3.1) shall be used to determine the font face.
- Character Directionality Override – When the contents of this run are displayed, this property acts as a right-to-left override for characters which are classified as follows (using the Unicode Character Database):
  - Weak types except European Number, European Number Terminator, Common Number Separator, Arabic Number and (for Hebrew text) European Number Separator when constituting part of a number
  - Neutral types
- [*Rationale*: This override allows applications to store and utilize higher-level information beyond that implicitly derived from the Unicode Bidirectional algorithm. For example, if the string "first second" appears in a right-to-left paragraph inside a document, the Unicode algorithm would always result in

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

"first second" at display time (since the neutral character is surrounded by strongly classified characters). However, if the whitespace was entered using a right-to-left input method (e.g. a Hebrew keyboard), then that character could be classified as RTL using this property, allowing the display of "second first" in a right-to-left paragraph, since the user explicitly asked for the space in a right-to-left context. *end rationale*]

This element provides information used to resolve the (Unicode) classifications of individual characters as either L, R, AN or EN. Once this is determined, the line should be displayed subject to the recommendation of the Unicode Bidirectional Algorithm in reordering resolved levels.

This property shall not be used with strong left-to-right text. Any behavior under that condition is unspecified. This property, when off, should not be used with strong right-to-left text. Any behavior under that condition is unspecified.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then right to left characteristics shall not be applied to the contents of this run.

[*Example*: Consider the following DrawingML visual content: "`first second,` أولى ثاني". This content might appear as follows within its parent paragraph:

```
<a:p>
  <a:r>
    <a:t>first second, </w:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t>أولى</a:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t> </a:t>
  </a:r>
  <a:r>
    <a:rPr>
      <a:rtl/>
    </a:rPr>
    <a:t>ثاني</a:t>
  </a:r>
</a:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The presence of the rtl element on the second, third, and fourth runs specifies that:

- The formatting on those runs is specified using the complex-script property variants.
- The whitespace character is treated as right-to-left.

Note that the second, third and fourth runs could be joined as one run with the rtl element specified.

end example]

Attributes	Description
val (On/Off Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property shall be explicitly turned off. [*Example*: For example, consider the following on/off property: `<… val="false"/>` The val attribute explicitly declares that the property is `false`. *end example*] The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.4.1. *end note*]

21.1.2.2.9 spcAft (Space After)

This element specifies the amount of vertical white space that is present after a paragraph. This space is specified in either percentage or points via the child elements spcPct and spcPts.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPts val="1800"/>
      </a:spcBef>
      <a:spcAft>
        <a:spcPts val="600"/>
      </a:spcAft>
    </a:pPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      …
      <a:t>Sample Text</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

The above paragraph of text is formatted to have a spacing both before and after the paragraph text. The spacing before is a size of 18 points, or value=1800 and the spacing after is a size of 6 points, or value=600. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

21.1.2.2.10 spcBef (Space Before)

This element specifies the amount of vertical white space that is present before a paragraph. This space is specified in either percentage or points via the child elements spcPct and spcPts.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPts val="1800"/>
      </a:spcBef>
      <a:spcAft>
        <a:spcPts val="600"/>
      </a:spcAft>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph of text is formatted to have a spacing both before and after the paragraph text. The spacing before is a size of 18 points, or value=1800 and the spacing after is a size of 6 points, or value=600. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacing) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.1.2.2.11 spcPct (Spacing Percent)

This element specifies the amount of white space that is to be used between lines and paragraphs in the form of a percentage of the text size. The text size that is used to calculate the spacing here is the text for each run, with the largest text size having precedence. That is if there is a run of text with 10 point font and within the same paragraph on the same line there is a run of text with a 12 point font size then the 12 point should be used to calculate the spacing to be used.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPct val="200%"/>
      </a:spcBef>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph of text is formatted to have a spacing before the paragraph text. This spacing is 200% of the size of the largest text on each line. *end example*]

Attributes	Description
val (Value)	Specifies the percentage of the size that the white space should be. The possible values for this attribute are defined by the ST_TextSpacingPercentOrPercentString simple type (§20.1.10.77).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacingPercent) is located in §A.4.1. *end note*]

21.1.2.2.12 spcPts (Spacing Points)

This element specifies the amount of white space that is to be used between lines and paragraphs in the form of a text point size. The size is specified using points where 100 is equal to 1 point font and 1200 is equal to 12 point.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:spcBef>
        <a:spcPts val="1400"/>
      </a:spcBef>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph of text is formatted to have a spacing before the paragraph text. This spacing is a size of 14 points due to val="1400". *end example*]

Attributes	Description
val (Value)	Specifies the size of the white space in point size. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. The possible values for this attribute are defined by the ST_TextSpacingPoint simple type (§20.1.10.78).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextSpacingPoint) is located in §A.4.1. *end note*]

21.1.2.2.13 tab (Tab Stop)

This element specifies a single tab stop to be used on a line of text when there are one or more tab characters present within the text. When there is more than one present than they should be utilized in increasing position order which is specified via the pos attribute.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:tabLst>
        <a:tab pos="2292350" algn="l"/>
        <a:tab pos="2627313" algn="l"/>
        <a:tab pos="2743200" algn="l"/>
        <a:tab pos="2974975" algn="l"/>
      </a:tabLst>
    </a:pPr>
    …
    <a:t>Sample Text</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The paragraph within which this <a:tab> information resides has a total of 4 unique tab stops that should be listed in order of increasing position. Along with specifying the tab position each tab allows for the specifying of an alignment. *end example*]

Attributes	Description
algn (Tab Alignment)	Specifies the alignment that is to be applied to text using this tab stop. If this attribute is omitted then the application default for the generating application. The possible values for this attribute are defined by the ST_TextTabAlignType simple type (§20.1.10.80).
pos (Tab Position)	Specifies the position of the tab stop relative to the left margin. If this attribute is omitted then the application default for tab stops is used. The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextTabStop) is located in §A.4.1. *end note*]

21.1.2.2.14 tabLst (Tab List)

This element specifies the list of all tab stops that are to be used within a paragraph. These tabs should be used when describing any custom tab stops within the document. If these are not specified then the default tab stops of the generating application should be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextTabStopList) is located in §A.4.1. *end note*]

21.1.2.3 Run Formatting

Run level formatting is the most granular property level and allows for the specifying of all low level text properties. The text run is what all paragraphs are derived from and thus specifying various properties per run allows for a diversely formatted text paragraph.

[*Example*: Consider the case where have multiple runs within a paragraph and you wish to apply bold to only one of them without having to split up the text into higher level XML groups. To do this we would simply apply the bold run property to the text run that we wish to format as shown below.

```
<a:r>
…
</a:r>
<a:r>
  <a:rPr lang="en-US" b="1" dirty="0" smtClean="0"/>
  <a:t>This text will be bold</a:t>
</a:r>
<a:r>
…
</a:r>
```

end example]

For each Unicode character in DrawingML text, the font face can be any of four font "slots": latin (§21.1.2.3.7), cs (§21.1.2.3.1), ea (§21.1.2.3.3), or sym (§21.1.2.3.10), as specified in the following table. For all ranges not explicitly called out below, the *ea* font shall be used.

Unicode Code Point Range	Classification
U+0000–U+007F	Use latin font
U+0080–U+00A6	Use latin font
U+00A9–U+00AF	Use latin font
U+00B2–U+00B3	Use latin font
U+00B5–U+00D6	Use latin font
U+00D8–U+00F6	Use latin font
U+00F8–U+058F	Use latin font
U+0590–U+074F	Use cs font
U+0780–U+07BF	Use cs font
U+0900–U+109F	Use cs font
U+10A0–U+10FF	Use latin font

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Unicode Code Point Range	Classification
U+1200–U+137F	Use latin font
U+13A0–U+177F	Use latin font
U+1D00–U+1D7F	Use latin font
U+1E00–U+1FFF	Use latin font
U+1780–U+18AF	Use cs font
U+2000–U+200B	Use latin font
U+200C–U+200F	Use cs font
U+2010–U+2029	Use latin font Except, for the quote characters in the range 2018 – 201F, use ea font if the text has one of the following language identifiers: ii-CN, ja-JP, ko-KR, zh-CN, zh-HK, zh-MO, zh-SG, zh-TW.
U+202A–U+202F	Use cs font
U+2030–U+2046	Use latin font
U+204A–U+245F	Use latin font
U+2670–U+2671	Use cs font
U+27C0–U+2BFF	Use latin font
U+3099–U+309A	Use ea font
U+D835	Use latin font [*Note*: This is the leading character for UTF-16 expression of Mathematical Alphanumeric characters (U+1D400 – U+1D7FF) and should be formatted using a latin font. *end note*]
U+F000–U+F0FF	Symbol, use sym font
U+FB00–U+FB17	Use latin font
U+FB1D–U+FB4F	Use cs font
U+FE50–U+FE6F	Use latin font
Otherwise	Use ea font

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.1.2.3.1 cs (Complex Script Font)

This element specifies that a complex script font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a complex script font.

[*Example*: Consider the DrawingML shown below.

©ISO/IEC 2016 – All rights reserved

```
<a:r>
  <a:rPr …>
    <a:cs typeface="Times New Roman"/>
  </a:rPr>
  <a:t>العربية </a:t>
</a:r>
```

The above run of text is rendered using the complex script font "Times New Roman". *end example*]

If the specified font is not available on a system being used for rendering, then the attributes of this element can be utilized to select an alternative font.

Attributes	Description
charset (Similar Character Set)	Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available. The value of this attribute shall be interpreted as follows:

Value	Description
0x00	Specifies the ANSI character set. (IANA name iso-8859-1)
0x01	Specifies the default character set.
0x02	Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.
0x4D	Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)
0x80	Specifies the JIS character set. (IANA name shift_jis)
0x81	Specifies the Hangul character set. (IANA name ks_c_5601-1987)
0x82	Specifies a Johab character set. (IANA name KS C-5601-1992)
0x86	Specifies the GB-2312 character set. (IANA name GBK)
0x88	Specifies the Chinese Big Five character set. (IANA name Big5)
0xA1	Specifies a Greek character set. (IANA name windows-1253)
0xA2	Specifies a Turkish character set. (IANA name iso-8859-9)
0xA3	Specifies a Vietnamese character set. (IANA name windows-1258)
0xB1	Specifies a Hebrew character set. (IANA name windows-1255)
0xB2	Specifies an Arabic character set. (IANA name windows-1256)
0xBA	Specifies a Baltic character set. (IANA name windows-1257)
0xCC	Specifies a Russian character set. (IANA name windows-1251)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>panose (Panose Setting)</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22.

The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</td><td>Specifies the font pitch as well as the font family for the corresponding font.

This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available.

The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41)datatype.</td></tr>
<tr><td>typeface (Text Typeface)</td><td>Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font.

The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.2 defRPr (Default Text Run Properties)

This element contains all default run level text properties for the text runs within a containing paragraph. These properties are to be used when overriding properties have not been defined within the rPr element.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<a:p>
  …
  <a:rPr u="sng"/>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

The run of text described above is formatting with a single underline of text matching color. *end example*]

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed.
	[*Example*: Consider the DrawingML shown below:
	`<a:p>`
	`…`
	`<a:rPr b="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	The above run of text is formatted as bold text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§20.1.10.64).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr i="1"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as italic text. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font. The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.73).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  <a:r>`
	`  <a:rPr lang="en-US" dirty="0" smtId="1"/>`
	`  <a:t>CNTS</a:t>`
	`  </a:r>`
	`  <a:endParaRPr lang="en-US" dirty="0"/>`
	`  </a:p>`
	`</p:txBody>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.74).

strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr strike="sngStrike"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single strikethrough text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.79).

sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> The above run of text is formatted with a 12 point text size. *end example*] The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.68).
u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed. [*Example*: Consider the DrawingML shown below. <a:p> … <a:rPr u="sng"/> … <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as single underline text. *end example*] The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.82).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.3.3 ea (East Asian Font)

This element specifies that an East Asian font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as an East Asian font.

[*Example*: Consider the DrawingML shown below.

```
<a:r>
  <a:rPr …>
    <a:ea typeface="SimSun"/>
  </a:rPr>
  <a:t>中文</a:t>
</a:r>
```

The above run of text is rendered using the East Asian font "SimSun". *end example*]

If the specified font is not available on a system being used for rendering, then the attributes of this element can be utilized to select an alternative font.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

The value of this attribute shall be interpreted as follows:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.</td></tr>
<tr><td>panose (Panose</td><td>Specifies the Panose-1 classification number for the current font using the mechanism</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Setting)	defined in §5.2.7.17 of ISO/IEC 14496-22. The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41)datatype.
typeface (Text Typeface)	Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.4 highlight (Highlight Color)

This element specifies the highlight color that is present for a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:highlight>
          <a:srgbClr val="FFFF00"/>
        </a:highlight>
      </a:rPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        …
        <a:t>Sample Text</a:t>
        …
      </a:r>
    </a:p>
    …
  </p:txBody>
```

The above run of text has a yellow highlight color as specified by the srgbClr child element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

21.1.2.3.5 hlinkClick (Click Hyperlink)

Specifies the on-click hyperlink information to be applied to a run of text. When the hyperlink text is clicked the link is fetched.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:hlinkClick r:id="rId2" tooltip="Some Sample Text"/>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The above run of text is a hyperlink that points to the resource pointed at by rId2 within this slides relationship file. Additionally this text should display a tooltip when the mouse is hovered over the run of text. *end example*]

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of 1 or `true` is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlink) is located in §A.4.1. *end note*]

21.1.2.3.6 hlinkMouseOver (Mouse-Over Hyperlink)

Specifies the mouse-over hyperlink information to be applied to a run of text. When the mouse is hovered over this hyperlink text the link is fetched.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:hlinkMouseOver r:id="rId2" tooltip="Some Sample Text"/>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The above run of text is a hyperlink that points to the resource pointed at by rId2 within this slides relationship file. Additionally this text should display a tooltip when the mouse is hovered over the run of text. *end example*]

Attributes	Description
action (Action Setting)	Specifies an action that is to be taken when this hyperlink is activated. This can be used to specify a slide to be navigated to or a script of code to be run. The possible values for this attribute are defined by the W3C XML Schema string datatype.
endSnd (End Sounds)	Specifies if the URL in question should stop all sounds that are playing when it is clicked. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
highlightClick (Highlight Click)	Specifies if this attribute has already been used within this document. That is when a hyperlink has already been visited that this attribute would be utilized so the generating application can determine the color of this text. If this attribute is omitted, then a value of 0 or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
history (Add Hyperlink to Page History)	Specifies whether to add this URI to the history when navigating to it. This allows for the viewing of this presentation without the storing of history information on the viewing machine. If this attribute is omitted, then a value of `1` or `true` is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Drawing Object Hyperlink Target) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship id that when looked up in this slides relationship file contains the target of this hyperlink. This attribute cannot be omitted. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
invalidUrl (Invalid URL)	Specifies the URL when it has been determined by the generating application that the URL is invalid. That is the generating application can still store the URL but it is known that this URL is not correct. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tgtFrame (Target Frame)	Specifies the target frame that is to be used when opening this hyperlink. When the hyperlink is activated this attribute is used to determine if a new window is launched for viewing or if an existing one can be used. If this attribute is omitted, than a new window is opened. The possible values for this attribute are defined by the W3C XML Schema string datatype.
tooltip (Hyperlink Tooltip)	Specifies the tooltip that should be displayed when the hyperlink text is hovered over with the mouse. If this attribute is omitted, than the hyperlink text itself can be displayed. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Hyperlink) is located in §A.4.1. *end note*]

21.1.2.3.7 latin (Latin Font)

This element specifies that a Latin font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a Latin font.

[*Example*: Consider the DrawingML shown below.

©ISO/IEC 2016 – All rights reserved

```
<a:r>
  <a:rPr …>
    <a:latin typeface="Sample Font"/>
  </a:rPr>
  <a:t>Sample Text</a:t>
</a:r>
```

The above run of text is rendered using the Latin font "Sample Font". *end example*]

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

The value of this attribute shall be interpreted as follows:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td></td><td>1250)</td></tr><tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr><tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr></table> The possible values for this attribute are defined by the W3C XML Schema byte datatype.
panose (Panose Setting)	Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22. The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41)datatype.
typeface (Text Typeface)	Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.8 r (Text Run)

This element specifies the presence of a run of text within the containing text body. The run element is the lowest level text separation mechanism within a text body. A text run can contain text run properties associated with the run. If no properties are listed then properties specified in the defRPr element are used.

[*Example*: Consider the case where the user would like to describe a text body that contains two runs of text and would like one to be bold and the other not. The following DrawingML would specify such a text body.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:txBody>
  …
  <a:r>
    <a:rPr b="1">
    </a:rPr>
    <a:t>Some text</a:t>
  </a:r>
  …
  <a:r>
    <a:rPr/>
    <a:t>Some text</a:t>
  </a:r>
</p:txBody>
```

The above text body has the first run be formatted bold and the second normally. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RegularTextRun) is located in §A.4.1. *end note*]

21.1.2.3.9 rPr (Text Run Properties)

This element contains all run level text properties for the text runs within a containing paragraph.

[*Example*: Consider the DrawingML shown below.

```
<a:p>
  …
  <a:rPr u="sng"/>
  …
  <a:t>Some Text</a:t>
  …
</a:p>
```

The run of text described above is formatting with a single underline of text matching color. *end example*]

Attributes	Description
altLang (Alternative Language)	Specifies the alternate language to use when the generating application is displaying the user interface controls. If this attribute is omitted, than the lang attribute is used here. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

b (Bold)	Specifies whether a run of text is formatted as bold text. If this attribute is omitted, than a value of 0, or false is assumed.
	[*Example*: Consider the DrawingML shown below.
	`<a:p>`
	`…`
	`<a:rPr b="1"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	… <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as bold text. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
baseline (Baseline)	Specifies the baseline for both the superscript and subscript fonts. The size is specified using a percentage where 1% is equal to 1 percent of the font size and 100% is equal to 100 percent font of the font size. The possible values for this attribute are defined by the ST_Percentage simple type (§20.1.10.40).
bmk (Bookmark Link Target)	Specifies the link target name that is used to reference to the proper link properties in a custom XML part within the document. The possible values for this attribute are defined by the W3C XML Schema string datatype.
cap (Capitalization)	Specifies the capitalization that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. The possible values for this attribute are defined by the ST_TextCapsType simple type (§20.1.10.64).
dirty (Dirty)	Specifies that the content of a text run has changed since the proofing tools have last been run. Effectively this flags text that is to be checked again by the generating application for mistakes such as spelling, grammar, etc. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
err (Spelling Error)	Specifies that when this run of text was checked for spelling, grammar, etc. that a mistake was indeed found. This allows the generating application to effectively save the state of the mistakes within the document instead of having to perform a full pass check upon opening the document. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
i (Italics)	Specifies whether a run of text is formatted as italic text. If this attribute is omitted, than a value of 0, or false is assumed. [*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:p> … <a:rPr i="1"/> … <a:t>Some Text</a:t> … </a:p> The above run of text is formatted as italic text. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
kern (Kerning)	Specifies the minimum font size at which character kerning occurs for this text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than kerning occurs for all font sizes down to a 0 point font. The possible values for this attribute are defined by the ST_TextNonNegativePoint simple type (§20.1.10.73).
kumimoji (Kumimoji)	Specifies whether the numbers contained within vertical text continue vertically with the text or whether they are to be displayed horizontally while the surrounding characters continue in a vertical fashion. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lang (Language ID)	Specifies the language to be used when the generating application is displaying the user interface controls. If this attribute is omitted, than the generating application can select a language of its choice. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).
noProof (No Proofing)	Specifies that a run of text has been selected by the user to not be checked for mistakes. Therefore if there are spelling, grammar, etc mistakes within this text the generating application should ignore them. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
normalizeH (Normalize Heights)	Specifies the normalization of height that is to be applied to the text run. This is a render-only modification and does not affect the actual characters stored in the text run. This attribute is also distinct from the toggle function where the actual characters stored in the text run are changed. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
smtClean (SmartTag Clean)	Specifies whether or not a text run has been checked for smart tags. This attribute acts much like the dirty attribute dose for the checking of spelling, grammar, etc. A value of true here indicates to the generating application that this text run should be checked for smart tags. If this attribute is omitted, than a value of 0, or false is assumed. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

smtId (SmartTag ID)	Specifies a smart tag identifier for a run of text. This ID is unique throughout the presentation and is used to reference corresponding auxiliary information about the smart tag. [*Note*: For a complete definition of smart tags, which are semantically identical throughout Office Open XML, see §17.5.1. *end note*]
	
	[*Example*: Consider the following DrawingML:
	
	`<p:txBody>`
	`  <a:bodyPr/>`
	`  <a:lstStyle/>`
	`  <a:p>`
	`  <a:r>`
	`  <a:rPr lang="en-US" dirty="0" smtId="1"/>`
	`  <a:t>CNTS</a:t>`
	`  </a:r>`
	`  <a:endParaRPr lang="en-US" dirty="0"/>`
	`  </a:p>`
	`</p:txBody>`
	
	The text run has a smtId attribute value of 1, which denotes that the text should be inspected for smart tag information, which in this case maps to a stock ticker symbol. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

spc (Spacing)	Specifies the spacing between characters within a text run. This spacing is specified numerically and should be consistently applied across the entire run of text by the generating application. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted than a value of 0 or no adjustment is assumed. The possible values for this attribute are defined by the ST_TextPoint simple type (§20.1.10.74).
strike (Strikethrough)	Specifies whether a run of text is formatted as strikethrough text. If this attribute is omitted, than no strikethrough is assumed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr strike="sngStrike"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted as single strikethrough text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextStrikeType simple type (§20.1.10.79).

sz (Font Size)	Specifies the size of text within a text run. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. If this attribute is omitted, than the value in defRPr should be used.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr sz="1200"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`
	
	The above run of text is formatted with a 12 point text size. *end example*]
	
	The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.68).

u (Underline)	Specifies whether a run of text is formatted as underlined text. If this attribute is omitted, than no underline is assumed.
	
	[*Example*: Consider the DrawingML shown below.
	
	`<a:p>`
	`…`
	`<a:rPr u="sng"/>`
	`…`
	`<a:t>Some Text</a:t>`
	`…`
	`</a:p>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The above run of text is formatted as single underline text. *end example*] The possible values for this attribute are defined by the ST_TextUnderlineType simple type (§20.1.10.82).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharacterProperties) is located in §A.4.1. *end note*]

21.1.2.3.10 sym (Symbol Font)

This element specifies that a symbol font be used for a specific run of text. This font is specified with a typeface attribute much like the others but is specifically classified as a symbol font.

[*Example*: Consider the DrawingML shown below.

```
<a:r>
  <a:rPr …>
    <a:sym typeface="Wingdings 2"/>
  </a:rPr>
  <a:t></a:t>
</a:r>
```

The above run of text, an example U+F02B (UTF-16) symbol from the Unicode Private Use Area, is rendered using the symbol font "Wingdings 2". *end example*]

Attributes	Description
charset (Similar Character Set)	Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available. The value of this attribute shall be interpreted as follows:

Value	Description
0x00	Specifies the ANSI character set. (IANA name iso-8859-1)
0x01	Specifies the default character set.
0x02	Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.
0x4D	Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>panose (Panose Setting)</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §5.2.7.17 of ISO/IEC 14496-22.

The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)</td><td>Specifies the font pitch as well as the font family for the corresponding font.

This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available.

The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41)datatype.</td></tr>
<tr><td>typeface (Text Typeface)</td><td>Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font.</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.3.11 t (Text String)

This element specifies the actual text for this text run. This is the text that is formatted using all specified body, paragraph and run properties. This element shall be present within a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    …
    <a:r>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
    …
  </a:p>
  …
</p:txBody>
```

The above DrawingML specifies a text body containing a single paragraph, containing a single run which contains the actual text specified with the <a:t> element.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.1.2.3.12 uFill (Underline Fill)

This element specifies the fill color of an underline for a run of text.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uFill>
          <a:solidFill>
            <a:srgbClr val="FFFF00"/>
          </a:solidFill>
        </a:uFill>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The underline color of the above text is yellow specified by the srgbClr child element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineFillGroupWrapper) is located in §A.4.1. *end note*]

21.1.2.3.13 uFillTx (Underline Fill Properties Follow Text)

This element specifies that the fill color of an underline for a run of text should be of the same color as the text run within which it is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uFillTx>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The underline color of the above text follows the color of the text run within which it resides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineFillFollowText) is located in §A.4.1. *end note*]

21.1.2.3.14 uLn (Underline Stroke)

This element specifies the properties for the stroke of the underline that is present within a run of text.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uLn algn="r">
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The underline alignment of the above text is right aligned. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.2.3.15 uLnTx (Underline Follows Text)

This element specifies that the stroke style of an underline for a run of text should be of the same as the text run within which it is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:r>
      <a:rPr …>
        <a:uLnTx>
      </a:rPr>
      …
      <a:t>Sample Text</a:t>
      …
    </a:r>
  </a:p>
  …
</p:txBody>
```

The underline stroke of the above text follows the stroke of the run text within which it resides. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextUnderlineLineFollowText) is located in §A.4.1. *end note*]

21.1.2.4 Bullets and Numbering

In addition to the above body, paragraph and text run properties there can also be a structure of bullets and numbering that can be defined by utilizing a few of these layers. Since Bullet and Numbering does span multiple formatting levels it is described on it's own in the following section.

21.1.2.4.1 buAutoNum (Auto-Numbered Bullet)

This element specifies that automatic numbered bullet points should be applied to a paragraph. These are not just numbers used as bullet points but instead automatically assigned numbers that are based on both buAutoNum attributes and paragraph level.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the DrawingML content shown below.

1. Bullet 1
    1. Bullet 2
2. Bullet 3

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buAutoNum type="arabicPeriod"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr lvl="1"…>
      <a:buAutoNum type="arabicPeriod"/>
    </a:pPr>
    …
    <a:t>Bullet 2</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr …>
      <a:buAutoNum type="arabicPeriod"/>
    </a:pPr>
    …
    <a:t>Bullet 3</a:t>
    …
  </a:p>
  …
</p:txBody>
```

For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Due to this breakdown of levels, the numbering sequence that should be automatically applied is 1, 1, 2 as is shown in the picture above. *end example*]

Attributes	Description
startAt (Start	Specifies the number that starts a given sequence of automatically numbered bullets.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

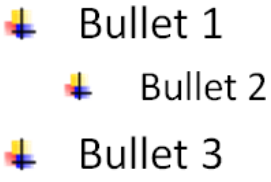
Attributes	Description
Numbering At)	When the numbering is alphabetical, the number should map to the appropriate letter. For instance 1 maps to 'a', 2 to 'b' and so on. If the numbers are larger than 26, then multiple letters should be used. For instance 27 should be represented as 'aa' and similarly 53 should be 'aaa'. The possible values for this attribute are defined by the ST_TextBulletStartAtNum simple type (§20.1.10.63).
type (Bullet Autonumbering Type)	Specifies the numbering scheme that is to be used. This allows for the describing of formats other than strictly numbers. For instance, a set of bullets can be represented by a series of Roman numerals instead of the standard 1,2,3,etc. number set. The possible values for this attribute are defined by the ST_TextAutonumberScheme simple type (§20.1.10.61).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextAutonumberBullet) is located in §A.4.1. *end note*]

21.1.2.4.2 buBlip (Picture Bullet)

This element specifies that a picture be applied to a set of bullets. This element allows for any standard picture format graphic to be used instead of the typical bullet characters. This opens up the possibility for bullets to be anything the generating application would seek to apply.

[*Example*: Consider the DrawingML shown below.




```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buBlip>
        <a:blip r:embed="rId2"/>
      </a:buBlip>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="1"…>
        <a:buBlip>
          <a:blip r:embed="rId2"/>
        </a:buBlip>
      </a:pPr>
      …
      <a:t>Bullet 2</a:t>
      …
    </a:p>
    <a:p>
      <a:pPr …>
        <a:buBlip>
          <a:blip r:embed="rId2"/>
        </a:buBlip>
      </a:pPr>
      …
      <a:t>Bullet 3</a:t>
      …
    </a:p>
    …
  </p:txBody>
```

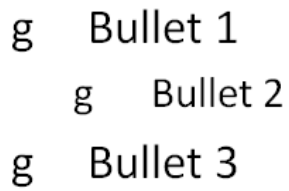
For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Because the same picture is specified for each bullet the levels do not stand out here. The only difference is the indentation as shown in the picture above. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBlipBullet) is located in §A.4.1. *end note*]

21.1.2.4.3 buChar (Character Bullet)

This element specifies that a character be applied to a set of bullets. These bullets are allowed to be any character in any font that the system is able to support. If no bullet font is specified along with this element then the paragraph font is used.

[*Example*: Consider the DrawingML shown below.



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFont typeface="Calibri"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr lvl="1"…>
      <a:buFont typeface="Calibri"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 2</a:t>
    …
  </a:p>
  <a:p>
    <a:pPr …>
      <a:buFont typeface="Calibri"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 3</a:t>
    …
  </a:p>
  …
</p:txBody>
```

For the above text there are a total of three bullet points. Two of which are at lvl="0" and one at lvl="1". Because the same character is specified for each bullet the levels do not stand out here. The only difference is the indentation as shown in the picture above. *end example*]

Attributes	Description
char (Bullet Character)	Specifies the character to be used in place of the standard bullet point. This character can be any character for the specified font that is supported by the system upon which this document is being viewed. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextCharBullet) is located in §A.4.1. *end note*]

21.1.2.4.4 buClr (Color Specified)

This element specifies the color to be used on bullet characters within a given paragraph. The color is specified using the numerical RGB color format.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buClr>
        <a:srgbClr val="FFFF00"/>
      </a:buClr>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The color of the above bullet does not follow the text color but instead has a yellow color specified by val="FFFF00". This color should only apply to the actual bullet character and not to the text within the bullet. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Color) is located in §A.4.1. *end note*]

21.1.2.4.5 buClrTx (Follow Text)

This element specifies that the color of the bullets for a paragraph should be of the same color as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buClrTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The color of the above bullet follows the default text color of the text for the run of text shown above since no specific text color was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletColorFollowText) is located in §A.4.1. *end note*]

21.1.2.4.6 buFont (Specified)

This element specifies the font to be used on bullet characters within a given paragraph. The font is specified using the typeface that it is registered as within the generating application.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFont typeface="Arial"/>
      <a:buChar char="g"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The font of the above bullet does not follow the text font but instead has Arial font specified by typeface="Arial". This font should only apply to the actual bullet character and not to the text within the bullet. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)</td><td>Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

The value of this attribute shall be interpreted as follows:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr>
<tr><td>0x01</td><td>Specifies the default character set.</td></tr>
<tr><td>0x02</td><td>Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.</td></tr>
<tr><td>0x4D</td><td>Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)</td></tr>
<tr><td>0x80</td><td>Specifies the JIS character set. (IANA name shift_jis)</td></tr>
<tr><td>0x81</td><td>Specifies the Hangul character set. (IANA name ks_c_5601-1987)</td></tr>
<tr><td>0x82</td><td>Specifies a Johab character set. (IANA name KS C-5601-1992)</td></tr>
<tr><td>0x86</td><td>Specifies the GB-2312 character set. (IANA name GBK)</td></tr>
<tr><td>0x88</td><td>Specifies the Chinese Big Five character set. (IANA name Big5)</td></tr>
<tr><td>0xA1</td><td>Specifies a Greek character set. (IANA name windows-1253)</td></tr>
<tr><td>0xA2</td><td>Specifies a Turkish character set. (IANA name iso-8859-9)</td></tr>
<tr><td>0xA3</td><td>Specifies a Vietnamese character set. (IANA name windows-1258)</td></tr>
<tr><td>0xB1</td><td>Specifies a Hebrew character set. (IANA name windows-1255)</td></tr>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.</td></tr>
<tr><td>panose (Panose</td><td>Specifies the Panose-1 classification number for the current font using the mechanism</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Setting)	defined in §5.2.7.17 of ISO/IEC 14496-22. The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. The possible values for this attribute are defined by the ST_PitchFamily simple type (§20.1.10.41)datatype.
typeface (Text Typeface)	Specifies the typeface, or name of the font that is to be used. The typeface is a string name of the specific font that should be used in rendering the presentation. If this font is not available within the font list of the generating application than font substitution logic should be utilized in order to select an alternate font. The possible values for this attribute are defined by the ST_TextTypeface simple type (§20.1.10.81).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

21.1.2.4.7 buFontTx (Follow text)

This element specifies that the font of the bullets for a paragraph should be of the same font as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buFontTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The font of the above bullet follows the default text font of the text for the run of text shown above since no specific text font was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletTypefaceFollowText) is located in §A.4.1. *end note*]

21.1.2.4.8 buNone (No Bullet)

This element specifies that the paragraph within which it is applied is to have no bullet formatting applied to it. That is to say that there should be no bulleting found within the paragraph where this element is specified.

[*Example*: Consider the DrawingML shown below.

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buNone/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The above paragraph is formatted with no bullets. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextNoBullet) is located in §A.4.1. *end note*]

21.1.2.4.9 buSzPct (Bullet Size Percentage)

This element specifies the size in percentage of the surrounding text to be used on bullet characters within a given paragraph.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzPct val="111%"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet follows the text size in that it is always rendered at 111% the size of the text within the given text run. This is specified by val="111%", with a restriction on the values not being less than 25% or more than 400%. This percentage size should only apply to the actual bullet character and not to the text within the bullet. *end example*]

Attributes	Description
val (Value)	Specifies the percentage of the text size that this bullet should be. This attribute should not be lower than 25% and not be higher than 400%. The possible values for this attribute are defined by the ST_TextBulletSizePercent simple type (§20.1.10.62).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizePercent) is located in §A.4.1. *end note*]

21.1.2.4.10 buSzPts (Bullet Size Points)

This element specifies the size in points to be used on bullet characters within a given paragraph. The size is specified using the points where 100 is equal to 1 point font and 1200 is equal to 12 point font.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzPts val="1400"/>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet does not follow the text size of the text within the given text run. The bullets size is specified by val="1400", which corresponds to a point size of 14. This bullet size should only apply to the actual bullet character and not to the text within the bullet. *end example*]

Attributes	Description
val (Value)	Specifies the size of the bullets in point size. Whole points are specified in increments of 100 starting with 100 being a point size of 1. For instance a font point size of 12 would be 1200 and a font point size of 12.5 would be 1250. The possible values for this attribute are defined by the ST_TextFontSize simple type (§20.1.10.68).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizePoint) is located in §A.4.1. *end note*]

21.1.2.4.11 buSzTx (Bullet Size Follows Text)

This element specifies that the size of the bullets for a paragraph should be of the same point size as the text run within which each bullet is contained.

[*Example*: Consider the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:p>
    <a:pPr …>
      <a:buSzTx>
    </a:pPr>
    …
    <a:t>Bullet 1</a:t>
    …
  </a:p>
  …
</p:txBody>
```

The size of the above bullet follows the default text size of the text for the run of text shown above since no specific text size was specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBulletSizeFollowText) is located in §A.4.1. *end note*]

21.1.2.4.12 lstStyle (Text List Styles)

This element specifies the list of styles associated with this body of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextListStyle) is located in §A.4.1. *end note*]

21.1.2.4.13 lvl1pPr (List Level 1 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="0". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl1pPr and thus create a paragraph of text that has no bullets and is right aligned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl1pPr algn="r">
      <a:buNone/>
    </a:lvl1pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="0">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<p:txBody> <a:bodyPr numCol="2" spcCol="914400"…/> <a:normAutofit/> </a:bodyPr> … <a:p> <a:pPr marL="0" algn="just"> <a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

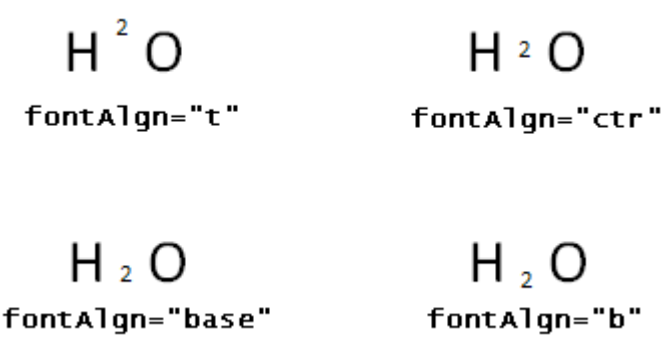
	without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr eaLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`
	`</a:r>`
	`…`
	`</p:txBody>`
	end example]
	
	The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).

hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	attribute is omitted, then a value of -342900 is implied. Here is some text Sample text Sample text Sample

Attributes	Description

	be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Test تجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:t>تجربة</a:t> </a:r> </a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.14 lvl2pPr (List Level 2 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="1". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl2pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl2pPr algn="r">
      <a:buNone/>
    </a:lvl2pPr>
  </a:lstStyle>
```

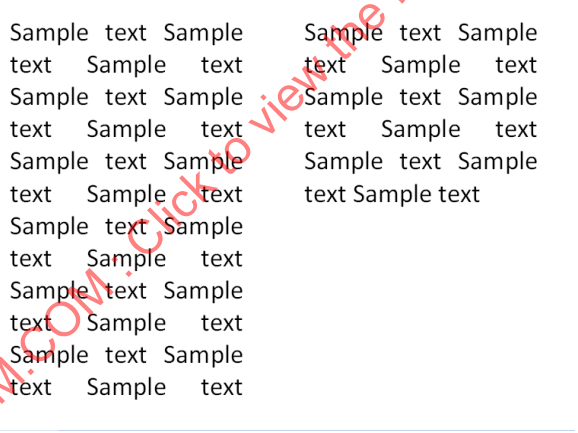
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="1">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

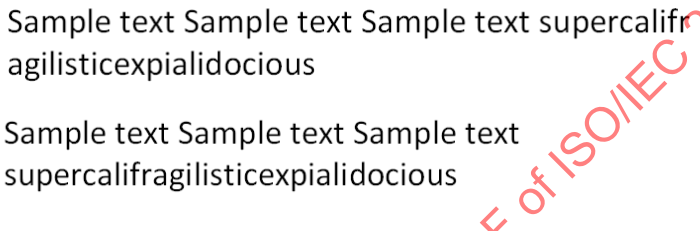
Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	 
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.15 lvl3pPr (List Level 3 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="2". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl3pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl3pPr algn="r">
      <a:buNone/>
    </a:lvl3pPr>
  </a:lstStyle>
```

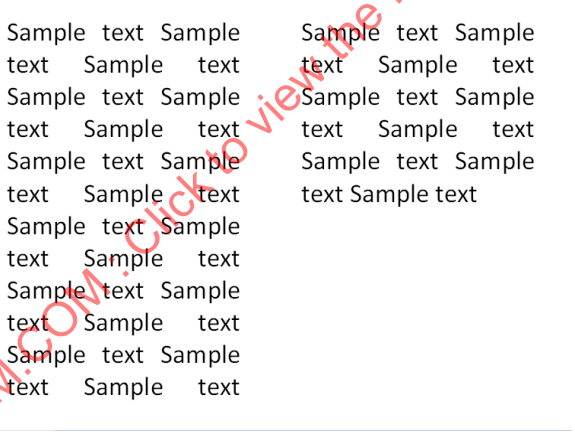
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="2">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.16 lvl4pPr (List Level 4 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="3". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl4pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl4pPr algn="r">
      <a:buNone/>
    </a:lvl4pPr>
  </a:lstStyle>
```

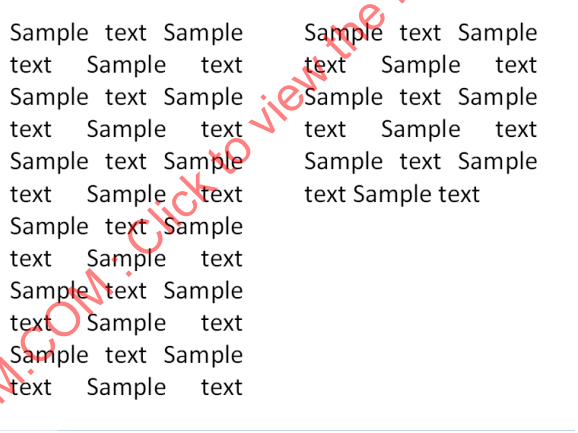
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="3">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.17 lvl5pPr (List Level 5 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="4". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl5pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl5pPr algn="r">
      <a:buNone/>
    </a:lvl5pPr>
  </a:lstStyle>
```

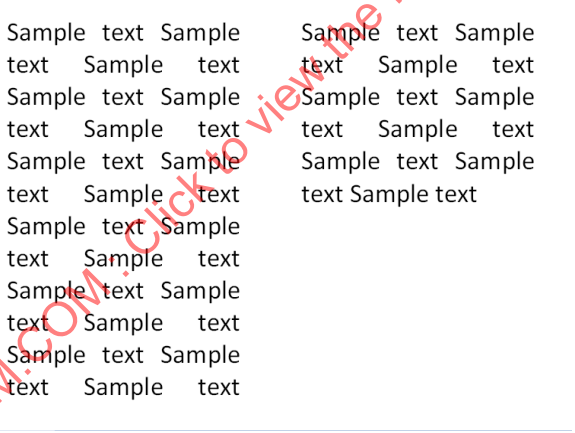
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="4">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr
	agilisticexpialidocious
	
	Sample text Sample text Sample text
	supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.18 lvl6pPr (List Level 6 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="5". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl6pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl6pPr algn="r">
      <a:buNone/>
    </a:lvl6pPr>
  </a:lstStyle>
```

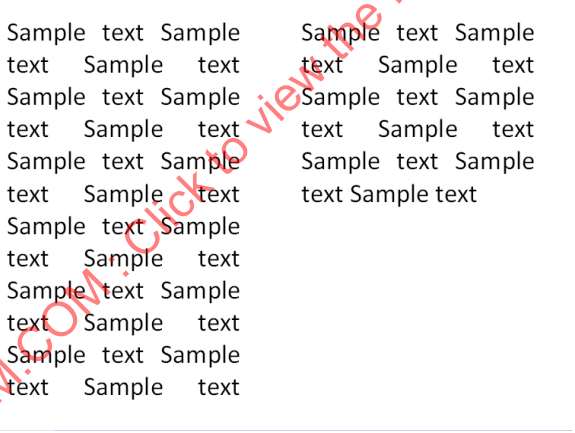
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="5">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr
	agilisticexpialidocious
	
	Sample text Sample text Sample text
	supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة</a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.19 lvl7pPr (List Level 7 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="6". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl7pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl7pPr algn="r">
      <a:buNone/>
    </a:lvl7pPr>
  </a:lstStyle>
```

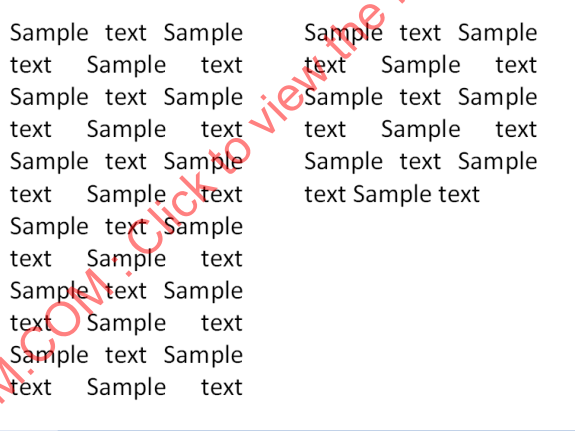
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <a:p>
      <a:pPr lvl="6">
      </a:pPr>
      …
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description
	<a:buNone/> </a:pPr> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

	long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	`fontAlgn="t"`
	
	H 2 O
	`fontAlgn="ctr"`
	
	H 2 O
	`fontAlgn="base"`
	
	H 2 O
	`fontAlgn="b"`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<a:txtBody> … <a:pPr fontAlgn="b" …/> … <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Here is some text Sample text Sample text Sample

Attributes	Description

	for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr latinLnBrk="0" …/>`
	`…`
	`<a:t>Sample text (Long word)</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr agilisticexpialidocious
	
	Sample text Sample text Sample text supercalifragilisticexpialidocious
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*] The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).
marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	Testتجربة ⟵ rtl = 0
	تجربةTest ⟵ rtl = 1
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:p> <a:p> <a:pPr rtl="1"/> <a:r> <a:rPr> <a:rtl w:val="0"/> </a:rPr> <a:t>Test </a:t> </a:r> <a:r> <a:t>تجربة </a:t> </a:r> </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.20 lvl8pPr (List Level 8 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="7". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl8pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl8pPr algn="r">
      <a:buNone/>
    </a:lvl8pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="7">
    </a:pPr>
    …
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <a:t>Some text</a:t>
      …
    </a:p>
  </p:txBody>
```

end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied. Sample text [*Example*: Consider the case where the user wishes to have two columns of text that have a justified alignment, much like text within a book. The following DrawingML could describe this. `<p:txBody>` `  <a:bodyPr numCol="2" spcCol="914400"…/>` `  <a:normAutofit/>` `  </a:bodyPr>` `  …` `  <a:p>` `  <a:pPr marL="0" algn="just">` `  <a:buNone/>` `  </a:pPr>` `  …` `  <a:t>Sample Text …</a:t>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	… </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextAlignType simple type (§20.1.10.59).
defTabSz (Default Tab Size)	Specifies the default size for a tab character within this paragraph. This attribute should be used to describe the spacing of tabs within the paragraph instead of a leading indentation tab. For indentation tabs there are the marL and indent attributes to assist with this. [*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this. <p:txBody> … <a:p> <a:pPr defTabSz="376300" …/> … <a:t>Sample Text …</a:t> … </a:p> </p:txBody> *end example*] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).
eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> …

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:p> <a:pPr eaLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied. [*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below. H 2 O fontAlgn="t" H 2 O fontAlgn="ctr" H 2 O fontAlgn="base" H 2 O fontAlgn="b" <a:txtBody> … <a:pPr fontAlgn="b" …/>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	… <a:r> <a:rPr …/> <a:t>H </a:t> </a:r> <a:r> <a:rPr sz="1200" …/> <a:t>2</a:t> </a:r> <a:r> <a:rPr …/> <a:t>O</a:t> </a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied. Here is some text Sample text Sample text Sample

Attributes	Description

	[*Example*: Consider the scenario where the user now wanted to add a paragraph indentation to the first line of text in their two column format book.
	
	`<p:txBody>`
	`  <a:bodyPr numCol="2" spcCol="914400"…/>`
	`  <a:normAutofit/>`
	`  </a:bodyPr>`
	`  …`
	`  <a:p>`
	`  <a:pPr marL="0" indent="571500" algn="just">`
	`  <a:buNone/>`
	`  </a:pPr>`
	`  …`
	`  <a:t>Here is some…</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*]
	
	The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.70).

latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr latinLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]
	
	The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).

marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).

rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1"/>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة </a:t>`
	`</a:r>`
	`</a:p>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</p:txBody> *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

21.1.2.4.21 lvl9pPr (List Level 9 Text Style)

This element specifies all paragraph level text properties for all elements that have the attribute lvl="8". There are a total of 9 level text property elements allowed, levels 0-8. It is recommended that the order in which this and other level property elements are specified be in order of increasing level. That is lvl2pPr should come before lvl3pPr. This allows the lower level properties to take precedence over the higher level ones because they are parsed first.

[*Example*: Consider the following DrawingML code that would specify a paragraph to follow the level style defined in lvl9pPr and thus create a paragraph of text that has no bullets and is right aligned.

```
<p:txBody>
  …
  <a:lstStyle>
    <a:lvl9pPr algn="r">
      <a:buNone/>
    </a:lvl9pPr>
  </a:lstStyle>
  <a:p>
    <a:pPr lvl="8">
    </a:pPr>
    …
    <a:t>Some text</a:t>
    …
  </a:p>
</p:txBody>
```

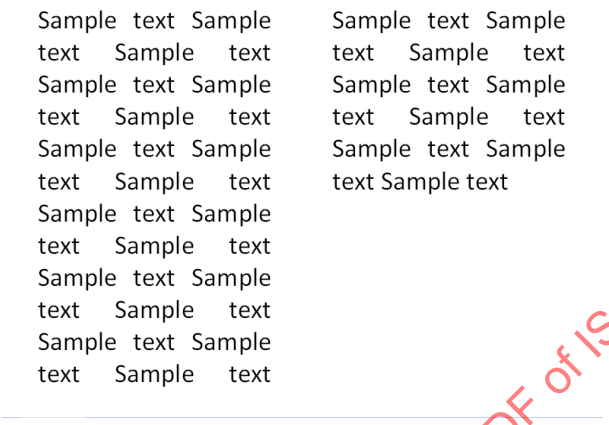
end example]

[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]

Attributes	Description
algn (Alignment)	Specifies the alignment that is to be applied to the paragraph. Possible values for this include left, right, centered, justified and distributed. If this attribute is omitted, then a value of left is implied.  Sample text Sample text Sample text

Attributes	Description

	with this.
	
	[*Example*: Consider the case where a paragraph contains numerous tabs that need to be of a specific size. The following DrawingML would describe this.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr defTabSz="376300" …/>`
	`  …`
	`  <a:t>Sample Text …</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

eaLnBrk (East Asian Line Break)	Specifies whether an East Asian word can be broken in half and wrapped onto the next line without a hyphen being added. To determine whether an East Asian word can be broken the presentation application would use the kinsoku settings here. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. That is it is not present within the existence of normal breakable East Asian words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of `1` or `true` is implied.
	
	[*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows.
	
	`<p:txBody>`
	`  …`
	`  <a:p>`
	`  <a:pPr eaLnBrk="0" …/>`
	`  …`
	`  <a:t>Sample text (Long word)</a:t>`
	`  …`
	`  </a:p>`
	`</p:txBody>`
	
	Sample text Sample text Sample text supercalifr
	agilisticexpialidocious

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fontAlgn (Font Alignment)	Determines where vertically on a line of text the actual words are positioned. This deals with vertical placement of the characters with respect to the baselines. For instance having text anchored to the top baseline, anchored to the bottom baseline, centered in between, etc. To understand this attribute and it's use it is helpful to understand what baselines are. A diagram describing these different cases is shown below. If this attribute is omitted, then a value of base is implied.
	
	[*Example*: Consider the case where the user wishes to represent the chemical compound of a water molecule. For this they need to make sure the H, the 2, and the O are all in the correct position and are of the correct size. The results below can be achieved through the DrawingML shown below.
	
	H 2 O
	fontAlgn="t"
	
	H 2 O
	fontAlgn="ctr"
	
	H 2 O
	fontAlgn="base"
	
	H 2 O
	fontAlgn="b"
	
	`<a:txtBody>`
	`…`
	`<a:pPr fontAlgn="b" …/>`
	`…`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>H </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr sz="1200" …/>`
	`<a:t>2</a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr …/>`
	`<a:t>O</a:t>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	</a:r> … </p:txBody> *end example*] The possible values for this attribute are defined by the ST_TextFontAlignType simple type (§20.1.10.66).
hangingPunct (Hanging Punctuation)	Specifies whether punctuation is to be forcefully laid out on a line of text or put on a different line of text. That is, if there is punctuation at the end of a run of text that should be carried over to a separate line does it actually get carried over. A true value allows for hanging punctuation forcing the punctuation to not be carried over and a value of false allows the punctuation to be carried onto the next text line. If this attribute is omitted, then a value of 0, or false is implied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
indent (Indent)	Specifies the indent size that is applied to the first line of text in the paragraph. An indentation of 0 is considered to be at the same location as marL attribute. If this attribute is omitted, then a value of -342900 is implied. Here is some text Sample text Sample text Sample

Attributes	Description
	</a:pPr> … <a:t>Here is some…</a:t> … </a:p> </p:txBody> By adding the indent attribute the user has effectively added a first line indent to this paragraph of text. *end example*] The possible values for this attribute are defined by the ST_TextIndent simple type (§20.1.10.70).
latinLnBrk (Latin Line Break)	Specifies whether a Latin word can be broken in half and wrapped onto the next line without a hyphen being added. This attribute is to be used specifically when there is a word that cannot be broken into multiple pieces without a hyphen. It is not present within the existence of normal breakable Latin words but is when a special case word arises that should not be broken for a line break. If this attribute is omitted, then a value of 1 or true is implied. [*Example*: Consider the case where the presentation contains a long word that must not be divided with a line break. Instead it should be placed, in whole on a new line so that it can fit. The picture below shows a normal paragraph where a long word has been broken for a line break. The second picture shown below shows that same paragraph with the long word specified to not allow a line break. The resulting DrawingML is as follows. <p:txBody> … <a:p> <a:pPr latinLnBrk="0" …/> … <a:t>Sample text (Long word)</a:t> … </a:p> </p:txBody> Sample text Sample text Sample text supercalifr agilisticexpialidocious Sample text Sample text Sample text supercalifragilisticexpialidocious *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lvl (Level)	Specifies the particular level text properties that this paragraph follows. The value for this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	attribute is numerical and formats the text according to the corresponding level paragraph properties that are listed within the lstStyle element. Since there are nine separate level properties defined, this tag has an effective range of 0-8 = 9 available values.
	
	[*Example*: Consider the following DrawingML. This would specify that this paragraph should follow the lvl2pPr formatting style because once again lvl="1" is considered to be level 2.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:pPr lvl="1" …/>`
	`…`
	`<a:t>Sample text</a:t>`
	`…`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	[*Note*: To resolve conflicting paragraph properties the linear hierarchy of paragraph properties should be examined starting first with the pPr element. The rule here is that properties that are defined at a level closer to the actual text should take precedence. That is if there is a conflicting property between the pPr and lvl1pPr elements then the pPr property should take precedence because in the property hierarchy it is closer to the actual text being represented. *end note*]
	
	The possible values for this attribute are defined by the ST_TextIndentLevelType simple type (§20.1.10.71).

marL (Left Margin)	Specifies the left margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marL attributes are additive with respect to the text position. If this attribute is omitted, then a value of 347663 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
marR (Right Margin)	Specifies the right margin of the paragraph. This is specified in addition to the text body inset and applies only to this text paragraph. That is the text body inset and the marR attributes are additive with respect to the text position. If this attribute is omitted, then a value of 0 is implied. The possible values for this attribute are defined by the ST_TextMargin simple type (§20.1.10.72).
rtl (Right To Left)	Specifies whether the text is right-to-left or left-to-right in its flow direction. If this

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	attribute is omitted, then a value of 0, or left-to-right is implied.
	
	[*Example*: Consider the following example of a text body with two lines of text. In this example, both lines contain English and Arabic text, however, the second line has the `rtl` attribute set to true whereas the first line does not set the `rtl` attribute.
	
	`<p:txBody>`
	`…`
	`<a:p>`
	`<a:r>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="1"/>`
	`</a:rPr>`
	`<a:t>تجربة</a:t>`
	`</a:r>`
	`</a:p>`
	`<a:p>`
	`<a:pPr rtl="1">`
	`<a:r>`
	`<a:rPr>`
	`<a:rtl w:val="0"/>`
	`</a:rPr>`
	`<a:t>Test </a:t>`
	`</a:r>`
	`<a:r>`
	`<a:t>تجربة </a:t>`
	`</a:r>`
	`</a:p>`
	`</p:txBody>`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextParagraphProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.1.2.5 Font Substitution

If any DrawingML element references a font and an appropriate format of the font is not stored within the document, the process of finding a suitable alternative font is known as *font substitution*.

The following elements specify font information: buFont@panose (§21.1.2.4.6); cs@panose (§21.1.2.3.1); ea@panose (§21.1.2.3.3); font@panose (§**19.2.1.13**); latin@panose (§21.1.2.3.7); sym@panose (§21.1.2.3.10).

The exact algorithm which is used for font substitution is highly dependent on the characteristics which are most desirable when performing the substitution: similar appearance of each glyph (to maximize visual familiarity), similar physical characteristics (to minimize changes in line height and breaking), etc. ISO/IEC 29500 recommends that applications look for the closest match to the following attribute values (in descending priority) in order to determine a suitable alternative font: panose, charset, pitchFamily, and typeface; however, applications are free to apply higher-order logic in its place.

21.1.3 Tables

This section contains information regarding the definition of a table within DrawingML. The following image is an example table within DrawingML.

text	text	text	text	text
text	text	text	text	text
text	text	text	text	text
text	text	text	text	text
text	text	text	text	text

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.1.3.1 cell3D (Cell 3-D)

This element specifies a set of properties which dictate the 3-D appearance of a given cell in a table. Collectively, these properties are referred to as a cell 3-D. The application of these properties occurs on a per-cell basis in the table.

Attributes	Description
prstMaterial (Preset Material)	Specifies a material type which is used to define the material characteristics of the cell. The material properties, combined with the lighting characteristics of the scene in define the final look and feel of the 3-D appearance of the cell. The possible values for this attribute are defined by the ST_PresetMaterialType simple

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	type (§20.1.10.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cell3D) is located in §A.4.1. *end note*]

21.1.3.2 gridCol (Table Grid Column)

This element specifies the width of a given column within a table. For each column in a table, there is an associated table grid column defining the width of the column.

[*Example*: Consider the following example of a table grid containing widths defined for three table grid columns:

```
<a:tblGrid>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
</a:tblGrid>
```

end example]

Attributes	Description
w (Width)	The width of the column in EMUs. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCol) is located in §A.4.1. *end note*]

21.1.3.3 headers (Header Cell Reference)

This element specifies a reference, using a unique identifier, to a table header cell that is associated with the current table cell. The identifier representing the reference shall be stored on this element's val attribute and is used to reference the unique identifier value of a table header cell. The contents of the table header cell designated by a specific unique identifier shall be used as the table header information associated with the table cell that references that specific unique identifier.

If this element is omitted or the value of the header cell identifier cannot be resolved, no header cell shall be associated with the current table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in DrawingML as follows:

```
<a:tbl>
  …
  <a:tr>
    …
    <a:tc >
      …
    </a:tc>
    <a:tc id="HeaderA">
      …
      <a:p>
        <a:r>
          <a:t>A</a:t>
        </a:r>
      </a:p>
    </a:tc>
    …
  </a:tr>
  <a:tr>
    …
    <a:tc id="HeaderC">
      …
      <a:p>
        <a:r>
          <a:t>C</a:t>
        </a:r>
      </a:p>
    </a:tc>
    <a:tc>
      <a:p>
        <a:r>
          <a:t>x1</a:t>
        </a:r>
      </a:p>
      <a:tcPr>
        …
        <a:headers>
          <a:header val="HeaderA" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
          <a:header val="HeaderC" />
        </a:headers>
        …
      </a:tcPr>
    </a:tc>
    …
  </a:tr>
</a:tbl>
```

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example the table cell with the content value of x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

21.1.3.4 headers (Header Cells Associated With Table Cell)

This element specifies the list of header cells, as specified by children header elements, that provide header information associated with the current table cell. Each header cell shall specify a unique identifier, as specified by the use of the attribute id on the header cell tc element. This element is typically used to gather header information about data and sub header cells.

If this element is omitted or there exists no children header element, no header cell shall be associated with the given table cell.

[*Example*: Consider the following 3 x 3 table with four header cells with values A, B, C, and D and four data cells with values x1, x2, y1, and y2:

	A	B
C	x1	x2
D	y1	y2

Each of the data cells is associated with two header cells and can be represented in DrawingML as follows:

```
<a:tbl>
  …
  <a:tr>
    …
    <a:tc >
      …
    </a:tc>
    <a:tc id="HeaderA">
      …
      <a:p>
        <a:r>
          <a:t>A</a:t>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
          </a:r>
        </a:p>
      </a:tc>
      …
    </a:tr>
    <a:tr>
      …
      <a:tc id="HeaderC">
        …
        <a:p>
          <a:r>
            <a:t>C</a:t>
          </a:r>
        </a:p>
      </a:tc>
      <a:tc>
        <a:p>
          <a:r>
            <a:t>x1</a:t>
          </a:r>
        </a:p>
        <a:tcPr>
          …
          <a:headers>
            <a:header val="HeaderA" />
            <a:header val="HeaderC" />
          </a:headers>
          …
        </a:tcPr>
      </a:tc>
      …
    </a:tr>
  </a:tbl>
```

The headers element specifies the list of header cells associated with the table cell that has a value of x1. In this example x1 is associated with headers that have an id of HeaderA and HeaderC. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Headers) is located in §A.4.1. *end note*]

21.1.3.5 lnB (Bottom Border Line Properties)

This element defines the line properties associated with the bottom border of a given cell.

[*Example*: Consider the following example of a bottom border line properties element within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:lnB w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnB>
```

In this example, one can see that the bottom border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.6 lnBlToTr (Bottom-Left to Top-Right Border Line Properties)

This element defines the line properties associated with the diagonal line from the bottom left corner of the cell to the top right corner.

[*Example*: Consider the following example of a lnBlToTr within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<a:lnBlToTr w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnBlToTr >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.7 lnL (Left Border Line Properties)

This element defines the line properties associated with the left border of a cell

[*Example*: Consider the following example of a lnL within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:lnL w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnL >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.8 lnR (Right Border Line Properties)

This element defines the line properties associated with right border of a cell.

[*Example*: Consider the following example of a lnR within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<a:lnR w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnR >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.9 lnT (Top Border Line Properties)

This element defines the line properties associated with the top border of a cell.

[*Example*: Consider the following example of a lnT within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<a:lnT w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnT >
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.10 lnTlToBr (Top-Left to Bottom-Right Border Line Properties)

This element defines the line properties associated with the diagonal line from the top left corner of the cell to the bottom right corner.

[*Example*: Consider the following example of a lnTlToBr within DrawingML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<a:lnTlToBr w="38100" cap="flat" cmpd="sng" algn="ctr">
  <a:solidFill>
    <a:schemeClr val="accent2"/>
  </a:solidFill>
  <a:prstDash val="solid"/>
  <a:round/>
  <a:headEnd type="none" w="med" len="med"/>
  <a:tailEnd type="none" w="med" len="med"/>
</a:lnTlToBr>
```

In this example, one can see that the border line style defined with certain properties, such as a flat end line cap, a given width, head and tail end, color, etc. *end example*]

Attributes	Description
algn (Stroke Alignment)	Specifies the alignment to be used for the underline stroke. The possible values for this attribute are defined by the ST_PenAlignment simple type (§20.1.10.39).
cap (Line Ending Cap Type)	Specifies the ending caps that should be used for this line. [*Note*: Examples of cap types are rounded, flat, etc. *end note*] If this attribute is omitted, than a value of square is assumed. The possible values for this attribute are defined by the ST_LineCap simple type (§20.1.10.31).
cmpd (Compound Line Type)	Specifies the compound line type to be used for the underline stroke. If this attribute is omitted, then a value of sng is assumed. The possible values for this attribute are defined by the ST_CompoundLine simple type (§20.1.10.15).
w (Line Width)	Specifies the width to be used for the underline stroke. If this attribute is omitted, then a value of 0 is assumed. The possible values for this attribute are defined by the ST_LineWidth simple type (§20.1.10.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineProperties) is located in §A.4.1. *end note*]

21.1.3.11 tableStyle (Table Style)

This element specifies a particular table style. Fourteen elements make up the styling information of a given table style. These fourteen elements work together to provide visual formatting options for on/off states of the following toggles:

- First row on/off - Associated element: firstRow

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Last row on/off - Associated element: lastRow
- First column on/off - Associated element: firstCol
- Last column on/off - Associated element: lastCol
- Row banding on/off - Associated elements: band1H, band2H
- Column banding on/off - Associated elements: band1V, band2V

The formatting associated with the wholeTbl element defines the table formatting when all options are off. When an option is turned on, the formatting for that particular option is applied to the table. The four cell specific formatting options are enabled when overlapping options are toggled on. For example, when the first row, and first column formatting options are enabled, any formatting within the northwest cell is also be applied since that is the overlapping table cell when both first column and first row formatting options are on.

[*Example*: Consider the following partial example of a tblStyle within DrawingML:

```
<a:tblStyle styleId="{5940675A-B579-460E-94D1-54222C63F5DA}"
  styleName="No Style, Table Grid">
  <a:wholeTbl>
    <a:tcTxStyle>
      <a:fontRef idx="minor">
        <a:scrgbClr r="0" g="0" b="0"/>
      </a:fontRef>
      <a:schemeClr val="tx1"/>
    </a:tcTxStyle>
    <a:tcStyle>
      <a:tcBdr>
        <a:left>
          <a:ln w="12700" cmpd="sng">
            <a:solidFill>
              <a:schemeClr val="tx1"/>
            </a:solidFill>
          </a:ln>
        </a:left>
```

…right, top, bottom, insideH, insideV border information is defined just as the 'left' tag…

```
        </a:tcBdr>
        <a:fill>
          <a:noFill/>
        </a:fill>
      </a:tcStyle>
    </a:wholeTbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<a:band1H>
  <a:tcStyle>
    <a:tcBdr/>
  </a:tcStyle>
</a:band1H>
```

…band2H, band1V, band2V, firstCol, firstRow, lastCol, lastRow, neCell, nwCell, seCell, swCell tags are all defined just as the 'band1H' tag

```
</a:tblStyle>
```

In this example, one can get an idea for the definition of a table style in its entirety. The above defined table style creates a style with only 1pt line formatting applied to all of the cells in a table. Notice that the on/off toggle formatting (band1H, band2H, firstCol, etc) do not define any formatting and therefore have no effect to the table when toggled. *end example*]

Attributes	Description
styleId (Style ID)	Specifies a GUID identifying the table style in a unique manner. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
styleName (Name)	Specifies the name of the table style which can show up in the user interface identifying the style to a user. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableStyle) is located in §A.4.1. *end note*]

21.1.3.12 tableStyleId (Table Style ID)

This element defines the table style which is currently applied to the table by referencing the styleId attribute of the tableStyle element.

[*Example*: Consider the following example of a tableStyleId within DrawingML:

```
<a:tblPr firstRow="1" bandRow="1">
  <a:tableStyleId>{5940675A-B579-460E-94D1-54222C63F5DA}</a:tableStyleId>
</a:tblPr>
```

In this example, we see a reference to a table style being specified in the tableStyleId element. *end example*]

The possible values for this element are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Guid) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.1.3.13 tbl (Table)

This element is the root element for a table. Within this element is contained everything that one would need to define a table within DrawingML.

[*Example*: Consider the following example of a tbl within DrawingML:

```
<a:tbl>
  <a:tblPr firstRow="1" bandRow="1">
  …
  <a:tblPr>
  <a:tblGrid>
  …
  </a:tblGrid>
  <a:tr h="419100">
  …
  </a:tr>
</a:tbl>
```

In this example, we see can see the definition of a table within DrawingML. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Table) is located in §A.4.1. *end note*]

21.1.3.14 tblGrid (Table Grid)

This element defines a list of table column (§21.1.3.2) elements. There should be a table column (§21.1.3.2) element for every column held within the table.

[*Example*: Consider the following example of a tblGrid within DrawingML:

```
<a:tblGrid>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
  <a:gridCol w="1117600"/>
</a:tblGrid>
```

In this example, we have a tblGrid defined that holds three columns, therefore the table has three columns. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableGrid) is located in §A.4.1. *end note*]

21.1.3.15 tblPr (Table Properties)

This element defines the properties of a table on the whole. Within this element are many visual modifications that can be applied to the table.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following example of a tblPr within DrawingML:

```
<a:tblPr firstRow="1" bandRow="1">
  <a:tableStyleId>{5940675A-B579-460E-94D1-54222C63F5DA}</a:tableStyleId>
</a:tblPr>
```

In this example, we see that there is a link to a table style id (§21.1.3.12) which is defined elsewhere and that the first column formatting and banded row formatting has been enabled. The table style defines the formatting applied with the two formatting options enabled. *end example*]

Attributes	Description

bandCol (Banded Columns)	Enables or disables the banded column formatting for a table style. A value of1 or `true` enables the banded column formatting defined in the table style. The attribute defaults to off if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr bandCol="1">`
	`  <a:tableStyleId>{5940675A-B579-460E-94D1-`
	`  54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the banded column formatting is enabled for the table. When applied, the linked table style defines the formatting for banded columns. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

bandRow (Banded Rows)	Enables or disables the banded row formatting for a table style. A value of 1 or `true` enables the banded row formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr bandRow="1">`
	`  <a:tableStyleId>{5940675A-B579-460E-94D1-`
	`  54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the banded row formatting is enabled for the table. When applied, the linked table style defines the formatting for banded rows. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

firstCol (First	Enables or disables the first column formatting for a table style. A value of 1 or `true`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description

Column)	enables the first column formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr firstCol="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the first column formatting is enabled for the table. When applied, the linked table style defines the formatting for the first column. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

firstRow (First Row)	Enables or disables the first row formatting for a table style. A value of `1` or `true` enables the first row formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr firstRow="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the first row formatting is enabled for the table. When applied, the linked table style defines the formatting for the first row. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lastCol (Last Column)	Enables or disables the last column formatting for a table style. A value of `1` or `true` enables the last column formatting defined in the table style. The attribute defaults to `false` if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr lastCol="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the last column formatting is enabled for the table. When applied, the linked table style defines the formatting for the last column. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

lastRow (Last Row)	Enables or disables the last row formatting for a table style. A value of 1 or true enables the last row formatting defined in the table style. The attribute defaults to false if it is not specified.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr lastRow="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see the last row formatting is enabled for the table. When applied, the linked table style defines the formatting for the last row. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

rtl (Right-to-Left)	Defines enables the right-to-left settings of a table. If the value of rtl is 1 or true , then the table is laid out from the right-to-left rather than the default left-to-right.
	
	[*Example*: Consider the following run:
	
	`<a:tblPr rtl="1">`
	`<a:tableStyleId>{5940675A-B579-460E-94D1-`
	`54222C63F5DA}</a:tableStyleId>`
	`</a:tblPr>`
	
	In this example, we can see that the table is to be created in a right-to-left direction. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableProperties) is located in §A.4.1. *end note*]

21.1.3.16 tc (Table Cell)

This element defines a cell within the table. The table cell holds a text body that actually contains the data held within the cell along with the properties of the table cell which hold formatting options associated with the cell.

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following example of a tc within DrawingML:

```
<a:tc>
  <a:txBody>
    <a:bodyPr/>
    <a:lstStyle/>
    <a:p>
      <a:pPr marL="0" algn="ctr" rtl="0"/>
      <a:r>
        <a:rPr lang="en-US" dirty="0" smtClean="0"/>
        <a:t>data</a:t>
      </a:r>
      <a:endParaRPr lang="en-US" dirty="0"/>
    </a:p>
  </a:txBody>
  <a:tcPr/>
</a:tc>
```

In this example, we see a single cell in a table being defined with the default cell properties and a text body which contains the word "data". The text "data" is the only text in the cell. *end example*]

Attributes	Description

gridSpan (Grid Span)	Specifies the number of columns that a merged cell spans. This is used in combination with the hMerge attribute on other cells in order to specify the beginning cell of a horizontal merge.
	
	[*Example*: Consider the following example:
	
	`<a:tc gridSpan="3">`
	…
	`/a:tc>`
	`<a:tc hMerge="1">`
	…
	`/a:tc>`
	`<a:tc hMerge="1">`
	…
	`/a:tc>`
	
	In this example, we can define what looks like a single cell in the table as a group of three cells merged together. The merged cell spans three columns of the table. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema int datatype.

hMerge (Horizontal Merge)	When this attribute is set to `1` or `true` , then this table cell is to be merged with the previous horizontal table cell when the table is created. [*Example*: Consider the following example:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<a:tc hMerge="1"> … </a:tc> In this example, we see the hMerge attribute set to on which signifies that this cell is to be merged with the previous horizontal cell in the table. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Table Cell Identifier)	Specifies a unique identifier for the current table cell. This identifier shall be unique within the table, and is used to identify this table cell as a header cell for other cells within the table, using the headers child element. If this element is omitted, this table cell has no unique identifier. [*Example*: Consider a table cell defined as follows: <a:tc id="januarytwentynine"> … </a:tc> The value in the id specifies a unique identifer oif januarytwentynine. Other cells in the table are then able to reference this cell as a row or column header by referencing this ID. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
rowSpan (Row Span)	Specifies the number of rows that a merged cell spans. This is used in combination with the vMerge attribute on other cells in order to specify the beginning cell of a horizontal merge. [*Example*: Consider the following example: <a:tc rowSpan="3"> … /a:tc> <a:tc vMerge="1"> … /a:tc> <a:tc vMerge="1"> … /a:tc> In this example, we can define what looks like a single cell in the table as a group of three

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	cells merged together. The merged cell spans three rows of the table. *end example*] The possible values for this attribute are defined by the W3C XML Schema int datatype.

vMerge (Vertical Merge)	When this attribute is set to 1 or true , then this table cell is to be merged with the previous vertical table cell when the table is created.
	
	[*Example*: Consider the following example:
	
	`<a:tc vMerge="1">`
	`…`
	`/a:tc>`
	
	In this example, we see the vMerge attribute set to on which signifies that this cell is to be merged with the previous vertical cell in the table. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCell) is located in §A.4.1. *end note*]

21.1.3.17 tcPr (Table Cell Properties)

This element defines the formatting properties associated with a cell. The formatting options which are available to be adjusted range from the line types used for the borders to the cell fill to the margins associated with the layout of the text in the cell.

[*Example*: Consider the following example of a tcPr within DrawingML:

```
<a:tcPr marL="45720" marR="45720">
  <a:lnL w="38100" cap="flat" cmpd="sng" algn="ctr">
    <a:solidFill>
      <a:schemeClr val="accent2"/>
    </a:solidFill>
    <a:prstDash val="solid"/>
    <a:round/>
    <a:headEnd type="none" w="med" len="med"/>
    <a:tailEnd type="none" w="med" len="med"/>
  </a:lnL>
</a:tcPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

In this example, we have a solid line defined as the left border of the cell along with left and right margin adjustments being made from the default margins. *end example*]

Attributes	Description
anchor (Anchor)	Defines the alignment of the text vertically within the cell. [*Example*: Consider the following example: `<a:tcPr marL="45720" anchor="ctr"/>` In this example, the text in the cell is anchored to the center of the cell vertically. *end example*] The possible values for this attribute are defined by the ST_TextAnchoringType simple type (§20.1.10.60).
anchorCtr (Anchor Center)	When this attribute is `1` or `true` , it modifies the anchor attribute. This attribute center-aligns the text box itself which allows for text to be left aligned along the center of the cell for example. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

horzOverflow (Horizontal Overflow)	Specifies the clipping behavior of the cell. The two options here allow for the text to be clipped and out of view when outside of the bounds of the cell, or for the text to remain visible and overflow outside of the cell.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr horzOverflow="overflow">`
	`…`
	`</a:tcPr>`
	
	In this example, the text in the cell freely overflows outside of the cell boundaries and always remains visible. *end example*]
	
	The possible values for this attribute are defined by the ST_TextHorzOverflowType simple type (§20.1.10.69).

marB (Bottom Margin)	Specifies the bottom margin of the cell. The value specified in this attribute is the distance to offset from the bottom of the cell.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marB="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the bottom of the cell. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	example] The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marL (Left Margin)	This attribute specifies the left margin of the cell. The value specified in this attribute is the distance to offset from the left of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marL="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the left of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marR (Right Margin)	This attribute specifies the right margin of the cell. The value specified in this attribute is the distance to offset from the right of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marR="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the right of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

marT (Top Margin)	This attribute specifies the top margin of the cell. The value specified in this attribute is the distance to offset from the top of the cell in EMU's.
	
	[*Example*: Consider the following example:
	
	`<a:tcPr marT="45720" anchor="ctr">`
	`…`
	`</a:tcPr>`
	
	In this example, we have specified a value for the margin on the top of the cell. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Coordinate32 simple type (§20.1.10.17).

vert (Text Direction)	Defines the text direction within the cell.
	
	[*Example*: Consider the following example:
	`<a:tcPr vert="vert270">`
	…
	`</a:tcPr>`
	
	In this example, we have rotated the layout of the text 270 degrees so that it starts at the bottom of the cell and goes upward toward the top of the cell. *end example*]
	
	The possible values for this attribute are defined by the ST_TextVerticalType simple type (§20.1.10.83).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableCellProperties) is located in §A.4.1. *end note*]

21.1.3.18 tr (Table Row)

This element defines a row in a table. A row as defined in a table is simply a listing of table cells (§21.1.3.16). There is a table row element defined for every row in the table.

[*Example*: Consider the following example of a tr within DrawingML:

```
<a:tr h="774700">
  <a:tc>
    <a:txBody>
      <a:bodyPr/>
      <a:lstStyle/>
      <a:p>
        <a:endParaRPr lang="en-US" dirty="0"/>
      </a:p>
    </a:txBody>
    <a:tcPr/>
  </a:tc>
…
</a:tr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example, we see a table row defined with an example table cell (§21.1.3.16) defined within it. The height of the row has been specified and in real use, there is a table cell defined in this row for each grid column (§21.1.3.2) defined in the table. *end example*]

Attributes	Description
h (Height)	Defines the height of the row in the table. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TableRow) is located in §A.4.1. *end note*]

21.2 DrawingML - Charts

The chart namespace in DrawingML is for representing visualizations of numeric data with column charts, pie charts, scatter charts, or other types of charts.

21.2.1 Table of Contents

This subclause is informative.

21.2.2 Elements 3367
21.2.2.1 applyToEnd (Apply to End) 3368
21.2.2.2 applyToFront (Apply To Front) 3368
21.2.2.3 applyToSides (Apply To Sides) 3368
21.2.2.4 area3DChart (3D Area Charts) 3369
21.2.2.5 areaChart (Area Charts) 3369
21.2.2.6 auto (Automatic Category Axis) 3369
21.2.2.7 autoTitleDeleted (Auto Title Is Deleted) 3369
21.2.2.8 autoUpdate (Update Automatically) 3370
21.2.2.9 axId (Axis ID) 3370
21.2.2.10 axPos (Axis Position) 3371
21.2.2.11 backWall (Back Wall) 3371
21.2.2.12 backward (Backward) 3371
21.2.2.13 bandFmt (Band Format) 3371
21.2.2.14 bandFmts (Band Formats) 3372
21.2.2.15 bar3DChart (3D Bar Charts) 3372
21.2.2.16 barChart (Bar Charts) 3372
21.2.2.17 barDir (Bar Direction) 3372
21.2.2.18 baseTimeUnit (Base Time Unit) 3372
21.2.2.19 bubble3D (3D Bubble) 3373
21.2.2.20 bubbleChart (Bubble Charts) 3373
21.2.2.21 bubbleScale (Bubble Scale) 3373
21.2.2.22 bubbleSize (Bubble Size) 3373
21.2.2.23 builtInUnit (Built in Display Unit Value) 3374

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.24 cat (Category Axis Data) 3374
21.2.2.25 catAx (Category Axis Data) 3374
21.2.2.26 chart (Reference to Chart Part) 3374
21.2.2.27 chart (Chart) 3375
21.2.2.28 chartObject (Chart Object) 3375
21.2.2.29 chartSpace (Chart Space) 3375
21.2.2.30 clrMapOvr (Color Map Override) 3375
21.2.2.31 crossAx (Crossing Axis ID) 3377
21.2.2.32 crossBetween (Cross Between) 3378
21.2.2.33 crosses (Crosses) 3378
21.2.2.34 crossesAt (Crossing Value) 3378
21.2.2.35 custSplit (Custom Split) 3379
21.2.2.36 custUnit (Custom Display Unit) 3379
21.2.2.37 data (Data Cannot Be Changed) 3379
21.2.2.38 date1904 (1904 Date System) 3380
21.2.2.39 dateAx (Date Axis) 3380
21.2.2.40 delete (Delete) 3380
21.2.2.41 depthPercent (Depth Percent) 3381
21.2.2.42 dispBlanksAs (Display Blanks As) 3381
21.2.2.43 dispEq (Display Equation) 3381
21.2.2.44 dispRSqr (Display R Squared Value) 3382
21.2.2.45 dispUnits (Display Units) 3382
21.2.2.46 dispUnitsLbl (Display Units Label) 3382
21.2.2.47 dLbl (Data Label) 3383
21.2.2.48 dLblPos (Data Label Position) 3383
21.2.2.49 dLbls (Data Labels) 3383
21.2.2.50 doughnutChart (Doughnut Charts) 3383
21.2.2.51 downBars (Down Bars) 3383
21.2.2.52 dPt (Data Point) 3383
21.2.2.53 dropLines (Drop Lines) 3384
21.2.2.54 dTable (Data Table) 3384
21.2.2.55 errBars (Error Bars) 3384
21.2.2.56 errBarType (Error Bar Type) 3384
21.2.2.57 errDir (Error Bar Direction) 3384
21.2.2.58 errValType (Error Bar Value Type) 3385
21.2.2.59 evenFooter (Even Footer) 3385
21.2.2.60 evenHeader (Even Header) 3385
21.2.2.61 explosion (Explosion) 3386
21.2.2.62 ext (Extension) 3386
21.2.2.63 externalData (External Data Relationship) 3387
21.2.2.64 extLst (Chart Extensibility) 3387
21.2.2.65 f (Formula) 3387
21.2.2.66 firstFooter (First Footer) 3388
21.2.2.67 firstHeader (First Header) 3388
21.2.2.68 firstSliceAng (First Slice Angle) 3389
21.2.2.69 floor (Floor) 3389
21.2.2.70 fmtId (Format ID) 3389
21.2.2.71 formatCode (Format Code) 3390

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.72 formatting (Formatting) 3390
21.2.2.73 forward (Forward) 3390
21.2.2.74 gapDepth (Gap Depth) 3391
21.2.2.75 gapWidth (Gap Width) 3391
21.2.2.76 grouping (Grouping) 3391
21.2.2.77 grouping (Bar Grouping) 3392
21.2.2.78 h (Height) 3392
21.2.2.79 headerFooter (Header and Footer) 3392
21.2.2.80 hiLowLines (High Low Lines) 3393
21.2.2.81 hMode (Height Mode) 3393
21.2.2.82 holeSize (Hole Size) 3394
21.2.2.83 hPercent (Height Percent) 3394
21.2.2.84 idx (Index) 3394
21.2.2.85 intercept (Intercept) 3395
21.2.2.86 invertIfNegative (Invert if Negative) 3395
21.2.2.87 lang (Editing Language) 3396
21.2.2.88 layout (Layout) 3396
21.2.2.89 layoutTarget (Layout Target) 3396
21.2.2.90 lblAlgn (Label Alignment) 3396
21.2.2.91 lblOffset (Label Offset) 3397
21.2.2.92 leaderLines (Leader Lines) 3397
21.2.2.93 legend (Legend) 3397
21.2.2.94 legendEntry (Legend Entry) 3397
21.2.2.95 legendPos (Legend Position) 3397
21.2.2.96 line3DChart (3D Line Charts) 3398
21.2.2.97 lineChart (Line Charts) 3398
21.2.2.98 logBase (Logarithmic Base) 3398
21.2.2.99 lvl (Level) 3398
21.2.2.100 majorGridlines (Major Gridlines) 3398
21.2.2.101 majorTickMark (Major Tick Mark) 3399
21.2.2.102 majorTimeUnit (Major Time Unit) 3399
21.2.2.103 majorUnit (Major Unit) 3399
21.2.2.104 manualLayout (Manual Layout) 3399
21.2.2.105 marker (Show Marker) 3400
21.2.2.106 marker (Marker) 3400
21.2.2.107 max (Maximum) 3400
21.2.2.108 min (Minimum) 3401
21.2.2.109 minorGridlines (Minor Gridlines) 3401
21.2.2.110 minorTickMark (Minor Tick Mark) 3401
21.2.2.111 minorTimeUnit (Minor Time Unit) 3401
21.2.2.112 minorUnit (Minor Unit) 3402
21.2.2.113 minus (Minus) 3402
21.2.2.114 multiLvlStrCache (Multi Level String Cache) 3402
21.2.2.115 multiLvlStrRef (Multi Level String Reference) 3402
21.2.2.116 name (Trendline Name) 3402
21.2.2.117 name (Pivot Name) 3402
21.2.2.118 noEndCap (No End Cap) 3403
21.2.2.119 noMultiLvlLbl (No Multi-level Labels) 3403

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.120 numCache (Number Cache) 3404
21.2.2.121 numFmt (Number Format) 3404
21.2.2.122 numLit (Number Literal) 3404
21.2.2.123 numRef (Number Reference) 3404
21.2.2.124 oddFooter (Odd Footer) 3404
21.2.2.125 oddHeader (Odd Header) 3405
21.2.2.126 ofPieChart (Pie of Pie or Bar of Pie Charts) 3405
21.2.2.127 ofPieType (Pie of Pie or Bar of Pie Type) 3406
21.2.2.128 order (Order) 3406
21.2.2.129 order (Polynomial Trendline Order) 3406
21.2.2.130 orientation (Axis Orientation) 3406
21.2.2.131 overlap (Overlap) 3407
21.2.2.132 overlay (Overlay) 3407
21.2.2.133 pageMargins (Page Margins) 3408
21.2.2.134 pageSetup (Page Setup) 3408
21.2.2.135 period (Period) 3412
21.2.2.136 perspective (Perspective) 3412
21.2.2.137 pictureFormat (Picture Format) 3413
21.2.2.138 pictureOptions (Picture Options) 3413
21.2.2.139 pictureStackUnit (Picture Stack Unit) 3413
21.2.2.140 pie3DChart (3D Pie Charts) 3414
21.2.2.141 pieChart (Pie Charts) 3414
21.2.2.142 pivotFmt (Pivot Format) 3414
21.2.2.143 pivotFmts (Pivot Formats) 3414
21.2.2.144 pivotSource (Pivot Source) 3414
21.2.2.145 plotArea (Plot Area) 3414
21.2.2.146 plotVisOnly (Plot Visible Only) 3414
21.2.2.147 plus (Plus) 3415
21.2.2.148 printSettings (Print Settings) 3415
21.2.2.149 protection (Protection) 3415
21.2.2.150 pt (Numeric Point) 3415
21.2.2.151 pt (String Point) 3416
21.2.2.152 ptCount (Point Count) 3416
21.2.2.153 radarChart (Radar Charts) 3416
21.2.2.154 radarStyle (Radar Style) 3416
21.2.2.155 rAngAx (Right Angle Axes) 3417
21.2.2.156 rich (Rich Text) 3417
21.2.2.157 rotX (X Rotation) 3417
21.2.2.158 rotY (Y Rotation) 3418
21.2.2.159 roundedCorners (Rounded Corners) 3418
21.2.2.160 scaling (Scaling) 3418
21.2.2.161 scatterChart (Scatter Charts) 3419
21.2.2.162 scatterStyle (Scatter Style) 3419
21.2.2.163 secondPiePt (Second Pie Point) 3419
21.2.2.164 secondPieSize (Second Pie Size) 3419
21.2.2.165 selection (Selection) 3420
21.2.2.166 separator (Separator) 3420
21.2.2.167 ser (Scatter Chart Series) 3420

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.168 ser (Area Chart Series) 3420
21.2.2.169 ser (Radar Chart Series) 3420
21.2.2.170 ser (Bar Chart Series) 3421
21.2.2.171 ser (Line Chart Series) 3421
21.2.2.172 ser (Pie Chart Series) 3421
21.2.2.173 ser (Surface Chart Series) 3421
21.2.2.174 ser (Bubble Chart Series) 3421
21.2.2.175 serAx (Series Axis) 3421
21.2.2.176 serLines (Series Lines) 3421
21.2.2.177 shape (Shape) 3422
21.2.2.178 showBubbleSize (Show Bubble Size) 3422
21.2.2.179 showCatName (Show Category Name) 3422
21.2.2.180 showDLblsOverMax (Show Data Labels over Maximum) 3423
21.2.2.181 showHorzBorder (Show Horizontal Border) 3423
21.2.2.182 showKeys (Show Legend Keys) 3424
21.2.2.183 showLeaderLines (Show Leader Lines) 3424
21.2.2.184 showLegendKey (Show Legend Key) 3424
21.2.2.185 showNegBubbles (Show Negative Bubbles) 3425
21.2.2.186 showOutline (Show Outline Border) 3425
21.2.2.187 showPercent (Show Percent) 3426
21.2.2.188 showSerName (Show Series Name) 3426
21.2.2.189 showVal (Show Value) 3427
21.2.2.190 showVertBorder (Show Vertical Border) 3427
21.2.2.191 sideWall (Side Wall) 3427
21.2.2.192 size (Size) 3428
21.2.2.193 sizeRepresents (Size Represents) 3428
21.2.2.194 smooth (Smoothing) 3428
21.2.2.195 splitPos (Split Position) 3429
21.2.2.196 splitType (Split Type) 3429
21.2.2.197 spPr (Shape Properties) 3429
21.2.2.198 stockChart (Stock Charts) 3430
21.2.2.199 strCache (String Cache) 3430
21.2.2.200 strLit (String Literal) 3430
21.2.2.201 strRef (String Reference) 3430
21.2.2.202 style (Style) 3430
21.2.2.203 surface3DChart (3D Surface Charts) 3431
21.2.2.204 surfaceChart (Surface Charts) 3431
21.2.2.205 symbol (Symbol) 3431
21.2.2.206 thickness (Thickness) 3431
21.2.2.207 tickLblPos (Tick Label Position) 3432
21.2.2.208 tickLblSkip (Tick Label Skip) 3432
21.2.2.209 tickMarkSkip (Tick Mark Skip) 3432
21.2.2.210 title (Title) 3432
21.2.2.211 trendline (Trendlines) 3432
21.2.2.212 trendlineLbl (Trendline Label) 3433
21.2.2.213 trendlineType (Trendline Type) 3433
21.2.2.214 tx (Chart Text) 3433
21.2.2.215 tx (Series Text) 3433

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.216 txPr (Text Properties) ... 3433
21.2.2.217 upBars (Up Bars) ... 3433
21.2.2.218 upDownBars (Up/Down Bars) ... 3434
21.2.2.219 userInterface (User Interface) ... 3434
21.2.2.220 userShapes (User Shapes) ... 3434
21.2.2.221 userShapes (Reference to Chart Drawing Part) ... 3434
21.2.2.222 v (Numeric Value) ... 3435
21.2.2.223 v (Text Value) ... 3435
21.2.2.224 val (Values) ... 3435
21.2.2.225 val (Error Bar Value) ... 3435
21.2.2.226 valAx (Value Axis) ... 3435
21.2.2.227 varyColors (Vary Colors by Point) ... 3436
21.2.2.228 view3D (View In 3D) ... 3436
21.2.2.229 w (Width) ... 3436
21.2.2.230 wireframe (Wireframe) ... 3437
21.2.2.231 wMode (Width Mode) ... 3437
21.2.2.232 x (Left) ... 3437
21.2.2.233 xMode (Left Mode) ... 3438
21.2.2.234 xVal (X Values) ... 3438
21.2.2.235 y (Top) ... 3438
21.2.2.236 yMode (Top Mode) ... 3438
21.2.2.237 yVal (Y Values) ... 3439

21.2.3 Simple Types ... 3439

21.2.3.1 ST_AxisUnit (Axis Unit) ... 3439
21.2.3.2 ST_AxPos (Axis Position) ... 3439
21.2.3.3 ST_BarDir (Bar Direction) ... 3440
21.2.3.4 ST_BarGrouping (Bar Grouping) ... 3440
21.2.3.5 ST_BubbleScale (Bubble Scale) ... 3441
21.2.3.6 ST_BuiltInUnit (Built-In Unit) ... 3441
21.2.3.7 ST_CrossBetween (Cross Between) ... 3441
21.2.3.8 ST_Crosses (Crosses) ... 3442
21.2.3.9 ST_DepthPercent (Depth Percent) ... 3442
21.2.3.10 ST_DispBlanksAs (Display Blanks As) ... 3442
21.2.3.11 ST_DLblPos (Data Label Position) ... 3443
21.2.3.12 ST_ErrBarType (Error Bar Type) ... 3444
21.2.3.13 ST_ErrDir (Error Bar Direction) ... 3444
21.2.3.14 ST_ErrValType (Error Value Type) ... 3444
21.2.3.15 ST_FirstSliceAng (First Slice Angle) ... 3445
21.2.3.16 ST_GapAmount (Gap Amount) ... 3445
21.2.3.17 ST_Grouping (Grouping) ... 3445
21.2.3.18 ST_HoleSize (Hole Size) ... 3446
21.2.3.19 ST_HPercent (Height Percent) ... 3446
21.2.3.20 ST_LayoutMode (Layout Mode) ... 3446
21.2.3.21 ST_LayoutTarget (Layout Target) ... 3447
21.2.3.22 ST_LblAlgn (Label Alignment) ... 3447
21.2.3.23 ST_LblOffset (Label Offset) ... 3447
21.2.3.24 ST_LegendPos (Legend Position) ... 3448

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.3.25 ST_LogBase (Logarithmic Base) 3448
21.2.3.26 ST_MarkerSize (Marker Size) 3448
21.2.3.27 ST_MarkerStyle (Marker Style) 3449
21.2.3.28 ST_OfPieType (Pie of Pie or Bar of Pie Type) 3450
21.2.3.29 ST_Order (Order) 3450
21.2.3.30 ST_Orientation (Orientation) 3450
21.2.3.31 ST_Overlap (Overlap) 3451
21.2.3.32 ST_PageSetupOrientation (Printed Page Orientation) 3451
21.2.3.33 ST_Period (Period) 3451
21.2.3.34 ST_Perspective (Perspective) 3452
21.2.3.35 ST_PictureFormat (Picture Format) 3452
21.2.3.36 ST_PictureStackUnit (Picture Stack Unit) 3452
21.2.3.37 ST_RadarStyle (Radar Style) 3453
21.2.3.38 ST_RotX (X Rotation) 3453
21.2.3.39 ST_RotY (Y Rotation) 3454
21.2.3.40 ST_ScatterStyle (Scatter Style) 3454
21.2.3.41 ST_SecondPieSize (Second Pie Size) 3455
21.2.3.42 ST_Shape (Shape) 3455
21.2.3.43 ST_SizeRepresents (Size Represents) 3455
21.2.3.44 ST_Skip (Skip) 3456
21.2.3.45 ST_SplitType (Split Type) 3456
21.2.3.46 ST_Style (Style) 3457
21.2.3.47 ST_TickLblPos (Tick Label Position) 3466
21.2.3.48 ST_TickMark (Tick Mark) 3466
21.2.3.49 ST_TimeUnit (Time Unit) 3466
21.2.3.50 ST_TrendlineType (Trendline Type) 3467
21.2.3.51 ST_DepthPercentWithSymbol (Depth Percent with Symbol) 3467
21.2.3.52 ST_HPercentWithSymbol (Height Percent with Symbol) 3467
21.2.3.53 ST_GapAmountPercent (Gap Amount Percentage) 3468
21.2.3.54 ST_SecondPieSizePercent (Second Pie Size Percentage) 3468
21.2.3.55 ST_HoleSizePercent (Hole Size Percentage) 3468
21.2.3.56 ST_LblOffsetPercent (Label Offset Percentage) 3468
21.2.3.57 ST_OverlapPercent (Overlap Percentage) 3468
21.2.3.58 ST_BubbleScalePercent (Bubble Scale Percentage) 3468
21.2.3.59 ST_Thickness (Thickness Percentage) 3468
21.2.3.60 ST_ThicknessPercent (Thickness Percentage) 3469

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

End of informative text.

21.2.2 Elements

In DrawingML, charts define a visualization of numeric data. The definition includes where the data shall come from, a cache of the data, and how the data shall be represented graphically. Other DrawingML elements are reused to define aspects of the formatting of the visualization.

See the informative material in Annex L for a description and overview of the basic chart types and chart components.

©ISO/IEC 2016 – All rights reserved

21.2.2.1 applyToEnd (Apply to End)

This element specifies the picture shall be applied to the end of the point or series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.2 applyToFront (Apply To Front)

This element specifies the picture shall be applied to the front of the point or series.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.3 applyToSides (Apply To Sides)

This element specifies the picture shall be applied to the sides of the point or series.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.4 area3DChart (3D Area Charts)

This element specifies the 3-D area series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Area3DChart) is located in §A.5.1. *end note*]

21.2.2.5 areaChart (Area Charts)

This element specifies the 2-D area series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AreaChart) is located in §A.5.1. *end note*]

21.2.2.6 auto (Automatic Category Axis)

This element specifies that this axis is a date or text axis based on the data that is used for the axis labels, not a specific choice.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.7 autoTitleDeleted (Auto Title Is Deleted)

This element specifies the title shall not be shown for this chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.8 autoUpdate (Update Automatically)

This element specifies the external data is updated automatically when the document containing the chart is opened.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.9 axId (Axis ID)

When specified as a child element of valAx, dateAx, catAx, or serAx, this element specifies the identifier for the axis. When specified as a child element of a chart, this element specifies the identifier of an axis that defines the coordinate space of the chart.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.10 axPos (Axis Position)

This element specifies the position of the axis on the chart.

Attributes	Description
val (Axis Position Value)	Specifies the position of the axis on the chart. The possible values for this attribute are defined by the ST_AxPos simple type (§21.2.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxPos) is located in §A.5.1. *end note*]

21.2.2.11 backWall (Back Wall)

This element specifies the back wall of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

21.2.2.12 backward (Backward)

This element specifies the number of categories (or units on a scatter chart) that the trend line extends before the data for the series that is being trended. On scatter and non-scatter charts, the value shall be any non-negative value.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.13 bandFmt (Band Format)

This element specifies the formatting band of a surface chart.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_BandFmt) is located in §A.5.1. *end note*]

21.2.2.14 bandFmts (Band Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BandFmts) is located in §A.5.1. *end note*]

21.2.2.15 bar3DChart (3D Bar Charts)

This element contains the 3-D bar or column series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bar3DChart) is located in §A.5.1. *end note*]

21.2.2.16 barChart (Bar Charts)

This element contains the 2-D bar or column series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarChart) is located in §A.5.1. *end note*]

21.2.2.17 barDir (Bar Direction)

This element specifies whether the series form a bar (horizontal) chart or a column (vertical) chart

Attributes	Description
val (Bar Direction Value)	Specifies the direction of the series. The possible values for this attribute are defined by the ST_BarDir simple type (§21.2.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarDir) is located in §A.5.1. *end note*]

21.2.2.18 baseTimeUnit (Base Time Unit)

This element specifies the smallest time unit that is represented on the date axis.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.19 bubble3D (3D Bubble)

This element specifies that the bubbles have a 3-D effect applied to them.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.20 bubbleChart (Bubble Charts)

This element contains the bubble series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleChart) is located in §A.5.1. *end note*]

21.2.2.21 bubbleScale (Bubble Scale)

This element specifies the scale factor for the bubble chart. This element can be a percentage value from 0 to 300, corresponding to a percentage of the default size.

Attributes	Description
val (Bubble Scale Value)	Specifies how to scale bubbles on a bubble chart. The possible values for this attribute are defined by the ST_BubbleScale simple type (§21.2.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleScale) is located in §A.5.1. *end note*]

21.2.2.22 bubbleSize (Bubble Size)

This element specifies the data for the sizes of the bubbles on the bubble chart.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.23 builtInUnit (Built in Display Unit Value)

This element specifies the display unit is one of the built in values.

Attributes	Description
val (Built In Unit Value)	Specifies the display unit scaling applied to the axis. The possible values for this attribute are defined by the ST_BuiltInUnit simple type (§21.2.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BuiltInUnit) is located in §A.5.1. *end note*]

21.2.2.24 cat (Category Axis Data)

This element specifies the data used for the category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxDataSource) is located in §A.5.1. *end note*]

21.2.2.25 catAx (Category Axis Data)

This element specifies the category axis of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CatAx) is located in §A.5.1. *end note*]

21.2.2.26 chart (Reference to Chart Part)

This element specifies the chart.

Attributes	Description
id (Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship for this Chart or Chart Drawing part. The type of relationship needed is specified by the parent element. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelId) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.27 chart (Chart)

This element specifies the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Chart) is located in §A.5.1. *end note*]

21.2.2.28 chartObject (Chart Object)

This element specifies that the chart cannot be edited by the user

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.29 chartSpace (Chart Space)

This element specifies overall settings for a single chart, and is the root node for the chart part.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartSpace) is located in §A.5.1. *end note*]

21.2.2.30 clrMapOvr (Color Map Override)

This element represents color mapping information. It is used to override the applications color mapping if the user has selected keep source formatting after a copy-paste.

Attributes	Description
accent1 (Accent 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 1 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent2 (Accent 2)	Specifies a color defined which is associated as the accent 2 color.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Namespace: http://purl.oclc.org/ooxml/drawingml/main	The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent3 (Accent 3) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 3 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent4 (Accent 4) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 4 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent5 (Accent 5) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 5 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
accent6 (Accent 6) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the accent 6 color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg1 (Background 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	A color defined which is associated as the first background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
bg2 (Background 2) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the second background color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
folHlink (Followed Hyperlink) Namespace:	Specifies a color defined which is associated as the color for a followed hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
http://purl.oclc.org/ooxml/drawingml/main	
hlink (Hyperlink) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the color for a hyperlink. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx1 (Text 1) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the first text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).
tx2 (Text 2) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a color defined which is associated as the second text color. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§20.1.10.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorMapping) is located in §A.4.1. *end note*]

21.2.2.31 crossAx (Crossing Axis ID)

This element specifies the ID of axis that this axis crosses. For instance, a category axis might cross a value axis, and the category axis's crossAx would contain the ID of the value axis.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.32 crossBetween (Cross Between)

This element specifies whether the value axis crosses the category axis between categories.

If not specified, then the application should choose an appropriate behavior.

Attributes	Description
val (Cross Between Value)	Specifies whether the value axis crosses the category axis between categories or on categories. The possible values for this attribute are defined by the ST_CrossBetween simple type (§21.2.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_CrossBetween) is located in §A.5.1. *end note*]

21.2.2.33 crosses (Crosses)

This element specifies how this axis crosses the perpendicular axis.

Attributes	Description
val (Crosses Value)	Specifies where the axis crosses its perpendicular axis. The possible values for this attribute are defined by the ST_Crosses simple type (§21.2.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Crosses) is located in §A.5.1. *end note*]

21.2.2.34 crossesAt (Crossing Value)

This element specifies where on the axis the perpendicular axis crosses. The units are dependent on the type of axis.

When specified as a child element of valAx, the value is a decimal number on the value axis. When specified as a child element of dateAx, the date is defined as a integer number of days relative to the base date of the current date system. When specified as a child element of catAx, the value is an integer category number, starting with 1 as the first category.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.35 custSplit (Custom Split)

This element contains the custom split information for a pie-of-pie or bar-of-pie chart with a custom split.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CustSplit) is located in §A.5.1. *end note*]

21.2.2.36 custUnit (Custom Display Unit)

This element specifies a custom value for the display unit.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.37 data (Data Cannot Be Changed)

This element specifies that the user cannot change the choice of data used for the chart

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.38 date1904 (1904 Date System)

This element specifies that the chart uses the 1904 date system. If the 1904 date system is used, then all dates and times shall be specified as a decimal number of days since Dec. 31, 1903. If the 1904 date system is not used, then all dates and times shall be specified as a decimal number of days since Dec. 31, 1899.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.39 dateAx (Date Axis)

This element specifies a date axis for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DateAx) is located in §A.5.1. *end note*]

21.2.2.40 delete (Delete)

This element specifies that the chart element specified by its containing element shall be deleted from the chart.

This should be set to true if the application adds these elements by default even after the user has specified that they should be removed from the chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `0` or `false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.41 depthPercent (Depth Percent)

This element specifies the depth of a 3-D chart as a percentage of the chart width (between 20 and 2000 percent).

Attributes	Description
val (Depth Percent Value)	Specifies a percentage value for the property defined by the parent XML element. The possible values for this attribute are defined by the ST_DepthPercent simple type (§21.2.3.9).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DepthPercent) is located in §A.5.1. *end note*]

21.2.2.42 dispBlanksAs (Display Blanks As)

This element specifies how blank cells shall be plotted on a chart .

Attributes	Description
val (Display Blanks As Value)	Specifies how blank cells are plotted on the chart. The possible values for this attribute are defined by the ST_DispBlanksAs simple type (§21.2.3.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispBlanksAs) is located in §A.5.1. *end note*]

21.2.2.43 dispEq (Display Equation)

This element specifies that the equation for the trendline is displayed on the chart (in the same label as the R-squared value).

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.44 dispRSqr (Display R Squared Value)

This element specifies that the R-squared value of the trendline is displayed on the chart (in the same label as the equation).

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.45 dispUnits (Display Units)

This element specifies the scaling value of the display units for the value axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispUnits) is located in §A.5.1. *end note*]

21.2.2.46 dispUnitsLbl (Display Units Label)

This element specifies the display unit label for the value axis in the specified chart.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DispUnitsLbl) is located in §A.5.1. *end note*]

21.2.2.47 dLbl (Data Label)

This element specifies a data label.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLbl) is located in §A.5.1. *end note*]

21.2.2.48 dLblPos (Data Label Position)

This element specifies the position of the data label.

Attributes	Description
val (Data Label Position Value)	Specifies how the data label is positioned on the chart. The possible values for this attribute are defined by the ST_DLblPos simple type (§21.2.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLblPos) is located in §A.5.1. *end note*]

21.2.2.49 dLbls (Data Labels)

This element serves as a root element that specifies the settings for the data labels for an entire series or the entire chart. It contains child elements that specify the specific formatting and positioning settings.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DLbls) is located in §A.5.1. *end note*]

21.2.2.50 doughnutChart (Doughnut Charts)

This element contains the doughnut chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DoughnutChart) is located in §A.5.1. *end note*]

21.2.2.51 downBars (Down Bars)

This element specifies the down bars.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBar) is located in §A.5.1. *end note*]

21.2.2.52 dPt (Data Point)

This element specifies a single data point.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DPt) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.53 dropLines (Drop Lines)

This element specifies drop lines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.54 dTable (Data Table)

This element specifies a data table.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DTable) is located in §A.5.1. *end note*]

21.2.2.55 errBars (Error Bars)

This element specifies error bars. The errValType element controls whether the minus, plus, or val elements are used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrBars) is located in §A.5.1. *end note*]

21.2.2.56 errBarType (Error Bar Type)

This element specifies the style of the error bars - positive, negative, or both.

Attributes	Description
val (Error Bar Type Value)	Specifies the style of error bars. The possible values for this attribute are defined by the ST_ErrBarType simple type (§21.2.3.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrBarType) is located in §A.5.1. *end note*]

21.2.2.57 errDir (Error Bar Direction)

This element specifies the direction of the error bars.

Attributes	Description
val (Error Bar Direction Value)	Specifies the direction of the error bars. The possible values for this attribute are defined by the ST_ErrDir simple type (§21.2.3.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrDir) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.58 errValType (Error Bar Value Type)

This element specifies the type of values used to determine the length of the error bars.

Attributes	Description
val (Error Bar Type Value)	Specifies the type of values of the error bars. The possible values for this attribute are defined by the ST_ErrValType simple type (§21.2.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ErrValType) is located in §A.5.1. *end note*]

21.2.2.59 evenFooter (Even Footer)

This element specifies the footer to use on even numbered pages. (See §18.3.1.38 for more information.)

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.60 evenHeader (Even Header)

This element specifies the header to use on even numbered pages. (See §18.3.1.39 for more information.)

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.61 explosion (Explosion)

This element specifies the amount the data point shall be moved from the center of the pie.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.62 ext (Extension)

This element specifies an extension that is used for future extensions to the current version of DrawingML. This allows for the specifying of currently unknown elements in the future that are used for later versions of generating applications.

Attributes	Description
uri (Uniform Resource Identifier)	Specifies the URI, or uniform resource identifier that represents the data stored under this tag. The URI is used to identify the correct 'server' that can process the contents of this tag. The possible values for this attribute are defined by the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Extension) is located in §A.5.1. *end note*]

21.2.2.63 externalData (External Data Relationship)

This element specifies the relationship to the data for this chart.

The data can be linked, pointing to a spreadsheet in another file. Or, the data can be embedded, contained in a separate part within the same xml package containing the chart. In this case, it shall be stored as an embedded Spreadsheet object in Office Open XML format.

This is not used by a spreadsheet application as the spreadsheet application can maintain its own reference to the data in the spreadsheet via the formula `<f>` element.

Attributes	Description
id (Relationship Reference) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship for this chart. The relationship explicitly targeted by this attribute shall either be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/package, or http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExternalData) is located in §A.5.1. *end note*]

21.2.2.64 extLst (Chart Extensibility)

This element contains tags used for future extensibility of the file format.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ExtensionList) is located in §A.5.1. *end note*]

21.2.2.65 f (Formula)

This element specifies a reference to source of the data contained in this chart. This shall be used by the spreadsheet application only. A presentation, or word processing application should use the `externalData` element.

This reference is in the form of a book, sheet, and cell reference or a book, optional sheet, and defined name reference. This reference does not include the equals sign. (Cell references are defined in §18.17. Link and external references are described in detail within §18.17.2.3.) [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<c:cat>
  <c:strRef>
    <c:f>Sheet1!$A$1:$C$1</c:f>
    <c:strCache>
…
    </c:strCache>
  </c:strRef>
</c:cat>
```

The above example shows a formula reference used for the string cahce. In this case the series names, which are referenced by the formula element, are in cells A1, B1, and C1.

end example]

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.66 firstFooter (First Footer)

This element specifies the footer to use on the first page. (See §18.3.1.41 for more information.)

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.67 firstHeader (First Header)

This element specifies the header to use on the first page. (See §18.3.1.42 for more information.)

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.68 firstSliceAng (First Slice Angle)

This element specifies the angle of the first pie or doughnut chart slice, in degrees (clockwise from up).

Attributes	Description
val (First Slice Angle Value)	Specifies the angle of the first slice. The possible values for this attribute are defined by the ST_FirstSliceAng simple type (§21.2.3.15).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FirstSliceAng) is located in §A.5.1. *end note*]

21.2.2.69 floor (Floor)

This element specifies the floor of a 3D chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

21.2.2.70 fmtId (Format ID)

This element represents a pivot format ID. It serves as a link back to the correct pivotTable which in turn specifies a link that then defines which set of chart format rules apply to this chart.

This ID shall match the chartFormat element, `chart` attribute, described in §18.10.1.12 of the SpreadsheetML reference material. The chartFormat element also contains a `format` attribute which is used to index into the pivotFmts collection (§21.2.2.143).

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.71 formatCode (Format Code)

This element specifies a string representing the format code to apply. For more information see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.72 formatting (Formatting)

This element specifies that a user cannot change formatting on chart elements.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.73 forward (Forward)

This element specifies the number of categories (or units on a scatter chart) that the trendline extends after the data for the series that is being trended. On scatter and non-scatter charts, the value shall be any non-negative value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.74 gapDepth (Gap Depth)

This element specifies the space between bar or column clusters, as a percentage of the bar or column width.

Attributes	Description
val (Gap Size Value)	Specifies that the contents of this attribute contain a gap amount between 0% and 500%. The possible values for this attribute are defined by the ST_GapAmount simple type (§21.2.3.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GapAmount) is located in §A.5.1. *end note*]

21.2.2.75 gapWidth (Gap Width)

This element specifies the space between bar or column clusters, as a percentage of the bar or column width.

Attributes	Description
val (Gap Size Value)	Specifies that the contents of this attribute contain a gap amount between 0% and 500%. The possible values for this attribute are defined by the ST_GapAmount simple type (§21.2.3.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GapAmount) is located in §A.5.1. *end note*]

21.2.2.76 grouping (Grouping)

This element specifies the kind of grouping for a column, line, or area chart.

Attributes	Description
val (Grouping	Specifies the grouping value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
Value)	The possible values for this attribute are defined by the ST_Grouping simple type (§21.2.3.17).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Grouping) is located in §A.5.1. *end note*]

21.2.2.77 grouping (Bar Grouping)

This element specifies the kind of grouping for a bar chart.

Attributes	Description
val (Bar Grouping Value)	Specifies the bar grouping value. The possible values for this attribute are defined by the ST_BarGrouping simple type (§21.2.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarGrouping) is located in §A.5.1. *end note*]

21.2.2.78 h (Height)

This element specifies the height (if Height Mode is Factor) or bottom (if Height Mode is edge) of the chart element as a fraction of the height of the chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.79 headerFooter (Header and Footer)

This element specifies the headers and footers that shall be used when the chart is printed. (See §18.3.1.46 for more information.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
alignWithMargins (Align With Margins)	Specifies the header and footer should align with the left and right margins of the chart. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentFirst (Different First)	Specifies the header and footer are different for the first page. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
differentOddEven (Different Odd Even)	Specifies the header and footer are different on odd-numbered pages and even-numbered pages. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_HeaderFooter) is located in §A.5.1. *end note*]

21.2.2.80 hiLowLines (High Low Lines)

This element specifies the high-low lines for the series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.81 hMode (Height Mode)

This element specifies how to interpret the Height element for this manual layout.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.82 holeSize (Hole Size)

This element specifies the size of the hole in a doughnut chart group.

Attributes	Description
val (Hole Size Value)	Specifies that the contents of this attribute contain a hole size between 10% and 90% of the size of the plot area. The possible values for this attribute are defined by the ST_HoleSize simple type (§21.2.3.18).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HoleSize) is located in §A.5.1. *end note*]

21.2.2.83 hPercent (Height Percent)

This element specifies the height of a 3-D chart as a percentage of the chart width.

Attributes	Description
val (Height Percent Value)	Specifies that the contents of this attribute contain a height percent between 5% and 500%. The possible values for this attribute are defined by the ST_HPercent simple type (§21.2.3.19).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HPercent) is located in §A.5.1. *end note*]

21.2.2.84 idx (Index)

This element specifies the index of the containing element. This index shall determine which of the parent's children collection this element applies to.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.85 intercept (Intercept)

This element specifies the value where the trendline shall cross the y axis. This property shall be supported only when the trendline type is exp, linear, or poly.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.86 invertIfNegative (Invert if Negative)

This element specifies the parent element shall invert its colors if the value is negative.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.87 lang (Editing Language)

This element specifies the primary editing language which was use when this chart was last modified.

Attributes	Description
val (Language Code)	Specifies a language tag as defined by RFC 3066. See simple type for additional information. The possible values for this attribute are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextLanguageID) is located in §A.5.1. *end note*]

21.2.2.88 layout (Layout)

This element specifies how the chart element is placed on the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Layout) is located in §A.5.1. *end note*]

21.2.2.89 layoutTarget (Layout Target)

This element specifies whether to layout the plot area by its inside (not including axis and axis labels) or outside (including axis and axis labels).

Attributes	Description
val (Layout Target Value)	Specifies the layout target value. The possible values for this attribute are defined by the ST_LayoutTarget simple type (§21.2.3.21).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutTarget) is located in §A.5.1. *end note*]

21.2.2.90 lblAlgn (Label Alignment)

This element specifies the text alignment for the tick labels on the axis.

Attributes	Description
val (Label Alignment Value)	Specifies the label alignment. The possible values for this attribute are defined by the ST_LblAlgn simple type (§21.2.3.22).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_LblAlgn) is located in §A.5.1. *end note*]

21.2.2.91 lblOffset (Label Offset)

This element specifies the distance of labels from the axis.

Attributes	Description
val (Label Offset Value)	Specifies the distance of labels from the axis. Shall contain a percentage between 0% and 1000%. The possible values for this attribute are defined by the ST_LblOffset simple type (§21.2.3.23).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LblOffset) is located in §A.5.1. *end note*]

21.2.2.92 leaderLines (Leader Lines)

This element specifies the leader lines for data labels.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.93 legend (Legend)

This element specifies the legend.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Legend) is located in §A.5.1. *end note*]

21.2.2.94 legendEntry (Legend Entry)

This element specifies a legend entry.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegendEntry) is located in §A.5.1. *end note*]

21.2.2.95 legendPos (Legend Position)

This element specifies the position of the legend.

Attributes	Description
val (Legend Position Value)	Specifies the position of the legend. The possible values for this attribute are defined by the ST_LegendPos simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(§21.2.3.24).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LegendPos) is located in §A.5.1. *end note*]

21.2.2.96 line3DChart (3D Line Charts)

This element contains the 3-D line chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Line3DChart) is located in §A.5.1. *end note*]

21.2.2.97 lineChart (Line Charts)

This element contains the 2-D line chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineChart) is located in §A.5.1. *end note*]

21.2.2.98 logBase (Logarithmic Base)

This element specifies the logarithmic base for a logarithmic axis.

Attributes	Description
val (Logarithmic Base Value)	Specifies the logarithmic base for a logarithmic axis. Shall contain a floating point value greater than or equal to 2. The possible values for this attribute are defined by the ST_LogBase simple type (§21.2.3.25).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LogBase) is located in §A.5.1. *end note*]

21.2.2.99 lvl (Level)

This element specifies data for a single level of labels for a category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Lvl) is located in §A.5.1. *end note*]

21.2.2.100 majorGridlines (Major Gridlines)

This element specifies major gridlines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.101 majorTickMark (Major Tick Mark)

This element specifies the major tick marks.

Attributes	Description
val (Tick Mark Value)	Specifies the major tick mark position. The possible values for this attribute are defined by the ST_TickMark simple type (§21.2.3.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickMark) is located in §A.5.1. *end note*]

21.2.2.102 majorTimeUnit (Major Time Unit)

This element specifies the time unit for major tick marks.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.103 majorUnit (Major Unit)

This element specifies the distance between major ticks.

Attributes	Description
val (Major Unit Value)	Specifies the distance between major ticks. Shall contain a positive floating-point number. The possible values for this attribute are defined by the ST_AxisUnit simple type (§21.2.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxisUnit) is located in §A.5.1. *end note*]

21.2.2.104 manualLayout (Manual Layout)

This element specifies the exact position of a chart element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_ManualLayout) is located in §A.5.1. *end note*]

21.2.2.105 marker (Show Marker)

This element is a Boolean that, when `true`, specifies that the marker shall be shown.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.106 marker (Marker)

This element specifies a data marker.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.1. *end note*]

21.2.2.107 max (Maximum)

This element specifies the maximum value of the axis.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.108 min (Minimum)

This element specifies the minimum value of the axis.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.109 minorGridlines (Minor Gridlines)

This element specifies the minor gridlines.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

21.2.2.110 minorTickMark (Minor Tick Mark)

This element specifies the minor tick marks for the axis.

Attributes	Description
val (Tick Mark Value)	Specifies the minor tick mark position. The possible values for this attribute are defined by the ST_TickMark simple type (§21.2.3.48).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickMark) is located in §A.5.1. *end note*]

21.2.2.111 minorTimeUnit (Minor Time Unit)

This element specifies the time unit for the minor tick marks.

Attributes	Description
val (Time Unit Value)	Specifies the time unit for the tick marks. The possible values for this attribute are defined by the ST_TimeUnit simple type (§21.2.3.49).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TimeUnit) is located in §A.5.1. *end note*]

21.2.2.112 minorUnit (Minor Unit)

This element specifies the distance between minor tick marks.

Attributes	Description
val (Minor Unit Value)	Specifies the distance between minor tick marks. Shall contain a positive floating-point number. The possible values for this attribute are defined by the ST_AxisUnit simple type (§21.2.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxisUnit) is located in §A.5.1. *end note*]

21.2.2.113 minus (Minus)

This element specifies the error bar value in the negative direction. It shall be used only when the errValType is cust.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.114 multiLvlStrCache (Multi Level String Cache)

This element specifies the last data shown on the chart for a category axis.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MultiLvlStrData) is located in §A.5.1. *end note*]

21.2.2.115 multiLvlStrRef (Multi Level String Reference)

This element specifies a reference to data for the category axis with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_MultiLvlStrRef) is located in §A.5.1. *end note*]

21.2.2.116 name (Trendline Name)

This element specifies the name of the trendline.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.117 name (Pivot Name)

This element specifies the name of the pivot table to get the data for the chart from.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.118 noEndCap (No End Cap)

This element specifies an end cap is not drawn on the error bars.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.119 noMultiLvlLbl (No Multi-level Labels)

This element specifies the labels shall be shown as flat text. If this element is not included or is set to false, then the labels shall be drawn as a hierarchy.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.120 numCache (Number Cache)

This element specifies the last data shown on the chart for a series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumData) is located in §A.5.1. *end note*]

21.2.2.121 numFmt (Number Format)

This element specifies number formatting for the parent element.

Attributes	Description
formatCode (Number Format Code)	This element specifies a string representing the format code to apply. For more information see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
sourceLinked (Linked to Source)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumFmt) is located in §A.5.1. *end note*]

21.2.2.122 numLit (Number Literal)

This element specifies a set of numbers used for the parent element.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumData) is located in §A.5.1. *end note*]

21.2.2.123 numRef (Number Reference)

This element specifies a reference to numeric data with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumRef) is located in §A.5.1. *end note*]

21.2.2.124 oddFooter (Odd Footer)

This element specifies the footer to use on odd numbered pages. (See §18.3.1.57 for more information.)

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.125 oddHeader (Odd Header)

This element specifies the header to use on odd numbered pages. (See §18.3.1.58 for more information.)

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.126 ofPieChart (Pie of Pie or Bar of Pie Charts)

This element contains the pie of pie or bar of pie series on this chart. Only the first series shall be displayed. The splitType element shall determine whether the splitPos and custSplit elements apply.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfPieChart) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.127 ofPieType (Pie of Pie or Bar of Pie Type)

This element specifies whether this chart is pie of pie or bar of pie.

Attributes	Description
val (Pie of Pie or Bar of Pie Type Value)	Specifies the type of pie of pie or bar of pie chart. The possible values for this attribute are defined by the ST_OfPieType simple type (§21.2.3.28).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfPieType) is located in §A.5.1. *end note*]

21.2.2.128 order (Order)

This element specifies the order of the series in the collection. It is 0 based.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.129 order (Polynomial Trendline Order)

This element specifies the order of the polynomial trend line. It is ignored for other trend line types.

Attributes	Description
val (Order Value)	Specifies that the contents of this attribute contain an integer between 2 and 6. The possible values for this attribute are defined by the ST_Order simple type (§21.2.3.29).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Order) is located in §A.5.1. *end note*]

21.2.2.130 orientation (Axis Orientation)

This element specifies the stretching and stacking of the picture on the data point, series, wall, or floor.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Orientation Value)	Specifies the orientation of the axis. The possible values for this attribute are defined by the ST_Orientation simple type (§21.2.3.30).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Orientation) is located in §A.5.1. *end note*]

21.2.2.131 overlap (Overlap)

This element specifies how much bars and columns shall overlap on 2-D charts.

Attributes	Description
val (Overlap Value)	Specifies how much bars and columns shall overlap on 2-D charts. Shall contain a percentage between -100% and 100%. The possible values for this attribute are defined by the ST_Overlap simple type (§21.2.3.31).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Overlap) is located in §A.5.1. *end note*]

21.2.2.132 overlay (Overlay)

This element specifies that other chart elements shall be allowed to overlap this chart element.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.133 pageMargins (Page Margins)

This element specifies the page margins for a chart.

Attributes	Description
b (Bottom)	Specifies the bottom page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
footer (Footer)	Specifies the footer margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
header (Header)	Specifies the header margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
l (Left)	Specifies the left page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
r (Right)	Specifies the right page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.
t (Top)	Specifies the top page margin in inches. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageMargins) is located in §A.5.1. *end note*]

21.2.2.134 pageSetup (Page Setup)

This element defines the page setup for the chart.

[*Example*: The following example shows the pageSetup element for ISO A0 paper, printed in black and white, with graphics:

```
<pageSetup blackAndWhite="true" draft="false" paperHeight="1189mm"
paperWidth="841mm"/>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>blackAndWhite (Black and White)</td><td>Specifies the page shall print in black and white.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>copies (Copies)</td><td>Specifies the number of copies that shall be printed.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.</td></tr>
<tr><td>draft (Draft)</td><td>Specifies the page shall be printed in draft mode.

The possible values for this attribute are defined by the W3C XML Schema boolean datatype.</td></tr>
<tr><td>firstPageNumber (First Page Number)</td><td>Specifies the page number.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.</td></tr>
<tr><td>horizontalDpi (Horizontal DPI)</td><td>Specifies the horizontal resolution to print in dots per inch.

The possible values for this attribute are defined by the W3C XML Schema int datatype.</td></tr>
<tr><td>orientation (Orientation)</td><td>Specifies the orientation of the paper.

The possible values for this attribute are defined by the ST_PageSetupOrientation simple type (§21.2.3.32).</td></tr>
<tr><td>paperHeight (Paper Height)</td><td>Height of custom paper as a number followed by a unit identifier. [Example: <code>297mm</code>, <code>11in</code> end example]

When paperHeight and paperWidth are specified, paperSize shall be ignored.

The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).</td></tr>
<tr><td>paperSize (Paper Size)</td><td>Specifies the paper size according to the following table.
<table>
<tr><th>Paper Size</th><th>Width</th><th>Height</th><th>Value</th></tr>
<tr><td>Letter paper</td><td>8.5 in.</td><td>11 in.</td><td>1</td></tr>
<tr><td>Legal paper</td><td>8.5 in.</td><td>14 in.</td><td>5</td></tr>
<tr><td>Standard paper</td><td>10 in.</td><td>11 in.</td><td>45</td></tr>
<tr><td>Standard paper</td><td>10 in.</td><td>14 in.</td><td>16</td></tr>
<tr><td>Standard paper</td><td>11 in.</td><td>17 in.</td><td>17</td></tr>
<tr><td>Standard paper</td><td>15 in.</td><td>11 in.</td><td>46</td></tr>
<tr><td>Standard paper</td><td>9 in.</td><td>11 in.</td><td>44</td></tr>
</table></td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description			
	SuperA/SuperA/A4 paper	227 mm	356 mm	57
	A2 paper	420 mm	594 mm	66
	A3 paper	297 mm	420 mm	8
	A3 extra paper	322 mm	445 mm	63
	A3 extra transverse paper	322 mm	445 mm	68
	A3 transverse paper	297 mm	420 mm	67
	A4 paper	210 mm	297 mm	9
	A4 extra paper	236 mm	322 mm	53
	A4 plus paper	210 mm	330 mm	60
	A4 transverse paper	210 mm	297 mm	55
	A4 small paper	210 mm	297 mm	10
	A5 paper	148 mm	210 mm	11
	A5 extra paper	174 mm	235 mm	64
	A5 transverse paper	148 mm	210 mm	61
	SuperB/SuperB/A3 paper	305 mm	487 mm	58
	B4 paper	250 mm	353 mm	12
	B5 paper	176 mm	250 mm	13
	ISO B5 extra paper	201 mm	276 mm	65
	JIS B5 transverse paper	182 mm	257 mm	62
	C paper	17 in.	22 in.	24
	D paper	22 in.	34 in.	25
	#10 envelope	4.125 in.	9.5 in.	20
	#11 envelope	4.5 in.	10.375 in.	21
	#12 envelope	4.75 in.	11 in.	22
	#14 envelope	5 in.	11.5 in.	23
	#9 envelope	3.875 in.	8.875 in.	19
	B4 envelope	250 mm	353 mm	33
	B5 envelope	176 mm	250 mm	34
	B6 envelope	176 mm	125 mm	35
	C3 envelope	324 mm	458 mm	29
	C4 envelope	229 mm	324 mm	30

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

C5 envelope	162 mm	229 mm	28
C6 envelope	114 mm	162 mm	31
C65 envelope	114 mm	229 mm	32
DL envelope	110 mm	220 mm	27
Invite envelope	220 mm	220 mm	47
Italy envelope	110 mm	230 mm	36
Monarch envelope	3.875 in.	7.5 in.).	37
6 3/4 envelope	3.625 in.	6.5 in.	38
E paper	34 in.	44 in.	26
Executive paper	7.25 in.	10.5 in.	7
German legal fanfold	8.5 in.	13 in.	41
German standard fanfold	8.5 in.	12 in.	40
US standard fanfold	14.875 in.	11 in.	39
Folio paper	8.5 in.	13 in.	14
ISO B4	250 mm	353 mm	42
Japanese double postcard	200 mm	148 mm	43
Ledger paper	17 in.	11 in.	4
Legal extra paper	9.275 in.	15 in.	51
Letter extra paper	9.275 in.	12 in.	50
Letter extra transverse paper	9.275 in.	12 in.	56
Letter plus paper	8.5 in.	12.69 in.	59
Letter transverse paper	8.275 in.	11 in.	54
Letter small paper	8.5 in.	11 in.	2
Note paper	8.5 in.	11 in.	18
Quarto paper	215 mm	275 mm	15
Statement paper	5.5 in.	8.5 in.	6
Tabloid paper	11 in.	17 in.	3
Tabloid extra paper	11.69 in.	18 in.	52

When paperHeight and paperWidth are specified, paperSize should be ignored.

The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.
paperWidth (Paper Width)	Width of custom paper as a number followed by a unit identifier. [*Example*: `21cm`, `8.5in` *end example*] When paperHeight and paperWidth are specified, paperSize shall be ignored. The possible values for this attribute are defined by the ST_PositiveUniversalMeasure simple type (§22.9.2.12).
useFirstPageNumber (Use First Page Number)	Specifies to use the first page number instead of automatically generating a page number. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
verticalDpi (Vertical DPI)	Specifies the vertical resolution to print in dots per inch. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PageSetup) is located in §A.5.1. *end note*]

21.2.2.135 period (Period)

This element specifies the period of the trend line for a moving average trend line. It is ignored for other trend line variants.

Attributes	Description
val (Period Value)	Specifies the period of the trend line for a moving average trend line. Shall contain an integer between 2 and 255. The possible values for this attribute are defined by the ST_Period simple type (§21.2.3.33).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Period) is located in §A.5.1. *end note*]

21.2.2.136 perspective (Perspective)

This element specifies the field of view angle for the 3-D chart. This element is ignored if Right Angle Axes is true.

Attributes	Description
val (Perspective	Specifies the field of view angle for the 3-D chart. Shall contain an integer between 0

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
Value)	and 240, whose unit is one-half degrees. The possible values for this attribute are defined by the ST_Perspective simple type (§21.2.3.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Perspective) is located in §A.5.1. *end note*]

21.2.2.137 pictureFormat (Picture Format)

This element specifies the stretching and stacking of the picture on the data point, series, wall, or floor.

Attributes	Description
val (Picture Format Value)	Specifies the stretching and stacking of the picture. The possible values for this attribute are defined by the ST_PictureFormat simple type (§21.2.3.35).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureFormat) is located in §A.5.1. *end note*]

21.2.2.138 pictureOptions (Picture Options)

This element specifies the picture to be used on the data point, series, wall, or floor.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureOptions) is located in §A.5.1. *end note*]

21.2.2.139 pictureStackUnit (Picture Stack Unit)

This element specifies the unit for each picture on the chart. This element applies only if the Picture Format is Stack and Scale.

Attributes	Description
val (Picture Stack Unit)	Specifies the unit for each picture on the chart. Shall contain a floating point number. The possible values for this attribute are defined by the ST_PictureStackUnit simple type (§21.2.3.36).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureStackUnit) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.140 pie3DChart (3D Pie Charts)

This element contains the 3-D pie series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pie3DChart) is located in §A.5.1. *end note*]

21.2.2.141 pieChart (Pie Charts)

This element contains the 2-D pie series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PieChart) is located in §A.5.1. *end note*]

21.2.2.142 pivotFmt (Pivot Format)

This element contains a set of formatting to be applied to the chart that is based on a pivotTable.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFmt) is located in §A.5.1. *end note*]

21.2.2.143 pivotFmts (Pivot Formats)

This element contains a collection of formatting bands for a surface chart indexed from low to high.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotFmts) is located in §A.5.1. *end note*]

21.2.2.144 pivotSource (Pivot Source)

This element specifies the source pivot table for a pivot chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PivotSource) is located in §A.5.1. *end note*]

21.2.2.145 plotArea (Plot Area)

This element specifies the plot area of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PlotArea) is located in §A.5.1. *end note*]

21.2.2.146 plotVisOnly (Plot Visible Only)

This element specifies that only visible cells should be plotted on the chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	omitted. A value of `0` or `false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.147 plus (Plus)

This element specifies the error bar value in the positive direction. It shall be used only when the errValType is cust.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.148 printSettings (Print Settings)

This element specifies the print settings for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PrintSettings) is located in §A.5.1. *end note*]

21.2.2.149 protection (Protection)

This element specifies protection for the chart. If the chart is on a protected worksheet or chart sheet, then these settings shall control how a user is able to interact with the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Protection) is located in §A.5.1. *end note*]

21.2.2.150 pt (Numeric Point)

This element specifies data for a particular data point.

Attributes	Description
formatCode (Number Format)	A string representing the format code to apply. For more information see see the SpreadsheetML numFmt element's (§18.8.30) formatCode attribute. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).
idx (Index)	The index of the series in the collection The possible values for this attribute are defined by the W3C XML Schema unsignedInt

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumVal) is located in §A.5.1. *end note*]

21.2.2.151 pt (String Point)

This element specifies string data for a specific data point.

Attributes	Description
idx (Index)	A 0 based index into a set of points. Represents the data point number this data is for. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrVal) is located in §A.5.1. *end note*]

21.2.2.152 ptCount (Point Count)

This element contains the number of values in the cache.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.153 radarChart (Radar Charts)

This element contains the radar chart series on this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarChart) is located in §A.5.1. *end note*]

21.2.2.154 radarStyle (Radar Style)

This element specifies what type of radar chart shall be drawn.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Radar Style Value)	Specifies the style of the radar chart. The possible values for this attribute are defined by the ST_RadarStyle simple type (§21.2.3.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarStyle) is located in §A.5.1. *end note*]

21.2.2.155 rAngAx (Right Angle Axes)

This element specifies that the chart axes are at right angles, rather than drawn in perspective. Applies only to 3-D charts.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.156 rich (Rich Text)

This element contains a string with rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.2.2.157 rotX (X Rotation)

This element specifies the amount a 3-D chart shall be rotated in the X direction.

Attributes	Description
val (X Rotation Value)	Specifies the amount a 3-D chart shall be rotated in the X direction. Shall contain an integer between -90 and 90. The possible values for this attribute are defined by the ST_RotX simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	(§21.2.3.38).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RotX) is located in §A.5.1. *end note*]

21.2.2.158 rotY (Y Rotation)

This element specifies the amount a 3-D chart shall be rotated in the Y direction.

Attributes	Description
val (Y Rotation Value)	Specifies the amount a 3-D chart shall be rotated in the Y direction. Shall contain an integer between 0 and 360. The possible values for this attribute are defined by the ST_RotY simple type (§21.2.3.39).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RotY) is located in §A.5.1. *end note*]

21.2.2.159 roundedCorners (Rounded Corners)

This element specifies the chart area shall have rounded corners.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0` or `false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.160 scaling (Scaling)

This element contains additional axis settings.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scaling) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.161 scatterChart (Scatter Charts)

This element contains the scatter chart series for this chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterChart) is located in §A.5.1. *end note*]

21.2.2.162 scatterStyle (Scatter Style)

This element specifies the kind of lines for the scatter chart.

Attributes	Description
val (Scatter Style Value)	Specifies the style of the scatter chart. The possible values for this attribute are defined by the ST_ScatterStyle simple type (§21.2.3.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterStyle) is located in §A.5.1. *end note*]

21.2.2.163 secondPiePt (Second Pie Point)

This element specifies a data point that shall be drawn in the second pie or bar in a pie of pie or bar of pie chart.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain an integer number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnsignedInt) is located in §A.5.1. *end note*]

21.2.2.164 secondPieSize (Second Pie Size)

This element specifies the size of the second pie or bar of a pie of pie chart or a bar of pie chart.

Attributes	Description
val (Second Pie Size Value)	Specifies the second pie or bar of a pie of pie chart or a bar of pie chart, as a percentage of the size of the first pie. Shall contain a percentage between 5% and 200%. The possible values for this attribute are defined by the ST_SecondPieSize simple type (§21.2.3.41).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_SecondPieSize) is located in §A.5.1. *end note*]

21.2.2.165 selection (Selection)

This element specifies the chart elements are protected from selection.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.166 separator (Separator)

This element specifies text that shall be used to separate the parts of a data label. The default is a comma, except for pie charts showing only category name and percentage, when a line break shall be used instead.

The possible values for this element are defined by the W3C XML Schema string datatype.

21.2.2.167 ser (Scatter Chart Series)

This element specifies a series on a scatter chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ScatterSer) is located in §A.5.1. *end note*]

21.2.2.168 ser (Area Chart Series)

This element specifies a series on an area chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AreaSer) is located in §A.5.1. *end note*]

21.2.2.169 ser (Radar Chart Series)

This element specifies a series on a radar chart.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadarSer) is located in §A.5.1. *end note*]

21.2.2.170 ser (Bar Chart Series)

This element specifies a series on a bar chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarSer) is located in §A.5.1. *end note*]

21.2.2.171 ser (Line Chart Series)

This element specifies a series on a line chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LineSer) is located in §A.5.1. *end note*]

21.2.2.172 ser (Pie Chart Series)

This element specifies a series on a doughnut or pie chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PieSer) is located in §A.5.1. *end note*]

21.2.2.173 ser (Surface Chart Series)

This element specifies a series on a surface chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SurfaceSer) is located in §A.5.1. *end note*]

21.2.2.174 ser (Bubble Chart Series)

This element specifies a series on a bubble chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BubbleSer) is located in §A.5.1. *end note*]

21.2.2.175 serAx (Series Axis)

This element specifies a series axis for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SerAx) is located in §A.5.1. *end note*]

21.2.2.176 serLines (Series Lines)

This element specifies series lines for the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChartLines) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.177 shape (Shape)

This element specifies the shape of a series or a 3-D bar chart.

Attributes	Description
val (Shape Value)	Specifies the shape of the series. The possible values for this attribute are defined by the ST_Shape simple type (§21.2.3.42).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.5.1. *end note*]

21.2.2.178 showBubbleSize (Show Bubble Size)

This element specifies the bubble size shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.179 showCatName (Show Category Name)

This element specifies that the category name shall be shown in the data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.180 showDLblsOverMax (Show Data Labels over Maximum)

This element specifies data labels over the maximum of the chart shall be shown.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.181 showHorzBorder (Show Horizontal Border)

This element specifies the horizontal borders shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.182 showKeys (Show Legend Keys)

This element specifies the legend keys shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.183 showLeaderLines (Show Leader Lines)

This element specifies leader lines shall be shown for data labels.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.184 showLegendKey (Show Legend Key)

This element specifies legend keys shall be shown in data labels.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.185 showNegBubbles (Show Negative Bubbles)

This element specifies negative sized bubbles shall be shown on a bubble chart.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.186 showOutline (Show Outline Border)

This element specifies the outline shall be shown on a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.187 showPercent (Show Percent)

This element specifies that the percentage shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.188 showSerName (Show Series Name)

This element specifies that the series name shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.189 showVal (Show Value)

This element specifies that the value shall be shown in a data label.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.190 showVertBorder (Show Vertical Border)

This element specifies the vertical border shall be shown in a data table.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.191 sideWall (Side Wall)

This element specifies the side wall.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.192 size (Size)

This element specifies the size of the marker in points.

Attributes	Description
val (Marker Size Value)	Specifies the size of the marker in points. Shall contain an integer between 2 and 72. The possible values for this attribute are defined by the ST_MarkerSize simple type (§21.2.3.26).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkerSize) is located in §A.5.1. *end note*]

21.2.2.193 sizeRepresents (Size Represents)

This element specifies how the bubble size values are represented on the chart.

Attributes	Description
val (Size Represents Value)	Specifies how the bubble sizes represent the values. The possible values for this attribute are defined by the ST_SizeRepresents simple type (§21.2.3.43).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SizeRepresents) is located in §A.5.1. *end note*]

21.2.2.194 smooth (Smoothing)

This element specifies the line connecting the points on the chart shall be smoothed using Catmull-Rom splines.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.195 splitPos (Split Position)

This element specifies a value that shall be used to determine which data points are in the second pie or bar on a pie of pie or bar of pie chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.196 splitType (Split Type)

This element specifies how to determine which data points are in the second pie or bar on a pie of pie or bar of pie chart.

Attributes	Description
val (Split Type Value)	Specifies how to split the data points between the first pie and second pie or bar. The possible values for this attribute are defined by the ST_SplitType simple type (§21.2.3.45).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SplitType) is located in §A.5.1. *end note*]

21.2.2.197 spPr (Shape Properties)

This element specifies the formatting for the parent chart element. The custGeom, prstGeom, scene3d, and xfrm elements are not supported. The bwMode attribute is not supported.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.2.2.198 stockChart (Stock Charts)

This element contains the collection of stock chart series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StockChart) is located in §A.5.1. *end note*]

21.2.2.199 strCache (String Cache)

This element specifies the last string data used for a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrData) is located in §A.5.1. *end note*]

21.2.2.200 strLit (String Literal)

This element specifies a set of strings used for a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrData) is located in §A.5.1. *end note*]

21.2.2.201 strRef (String Reference)

This element specifies a reference to data for a single data label or title with a cache of the last values used.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StrRef) is located in §A.5.1. *end note*]

21.2.2.202 style (Style)

This element specifies the style that shall be applied to the chart.

Attributes	Description
val (Style Type)	Specifies the chart style. The possible values for this attribute are defined by the ST_Style simple type (§21.2.3.46).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Style) is located in §A.5.1. *end note*]

21.2.2.203 surface3DChart (3D Surface Charts)

This element contains the set of 3-D surface series.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Surface3DChart) is located in §A.5.1. *end note*]

21.2.2.204 surfaceChart (Surface Charts)

This element contains the set of 2-D contour charts.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SurfaceChart) is located in §A.5.1. *end note*]

21.2.2.205 symbol (Symbol)

This element specifies the marker that is used for the data points.

Attributes	Description
val (Marker Style Value)	Specifies the marker style. The possible values for this attribute are defined by the ST_MarkerStyle simple type (§21.2.3.27).

[*Note*: The W3C XML Schema definition of this element's content model (CT_MarkerStyle) is located in §A.5.1. *end note*]

21.2.2.206 thickness (Thickness)

This element specifies the thickness of the walls or floor as a percentage of the largest dimension of the plot volume.

Attributes	Description
val (Integer Value)	Specifies that the contents of this attribute contain a percentage. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_Thickness simple type (§21.2.3.59).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Thickness) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.2.207 tickLblPos (Tick Label Position)

This element specifies the position of the tick labels on the axis.

Attributes	Description
val (Tick Label Position Value)	Specifies the tick label position. The possible values for this attribute are defined by the ST_TickLblPos simple type (§21.2.3.47).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TickLblPos) is located in §A.5.1. *end note*]

21.2.2.208 tickLblSkip (Tick Label Skip)

This element specifies how many tick labels to skip between label that is drawn.

Attributes	Description
val (Tick Skip Value)	Specifies the how many tick labels to skip between label that is drawn. Shall contain an integer greater than or equal to one. The possible values for this attribute are defined by the ST_Skip simple type (§21.2.3.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Skip) is located in §A.5.1. *end note*]

21.2.2.209 tickMarkSkip (Tick Mark Skip)

This element specifies how many tick marks shall be skipped before the next one shall be drawn.

Attributes	Description
val (Tick Skip Value)	Specifies the how many tick marks shall be skipped before the next one shall be drawn. Shall contain an integer greater than or equal to one. The possible values for this attribute are defined by the ST_Skip simple type (§21.2.3.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Skip) is located in §A.5.1. *end note*]

21.2.2.210 title (Title)

This element specifies a title.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Title) is located in §A.5.1. *end note*]

21.2.2.211 trendline (Trendlines)

This element specifies a trendline.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Trendline) is located in §A.5.1. *end note*]

21.2.2.212 trendlineLbl (Trendline Label)

This element specifies the label for the trendline.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrendlineLbl) is located in §A.5.1. *end note*]

21.2.2.213 trendlineType (Trendline Type)

This element specifies the style of the trendline.

Attributes	Description
val (Trendline Type Value)	Specifies the trendline style. The possible values for this attribute are defined by the ST_TrendlineType simple type (§21.2.3.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TrendlineType) is located in §A.5.1. *end note*]

21.2.2.214 tx (Chart Text)

This element specifies text to use on a chart, including rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Tx) is located in §A.5.1. *end note*]

21.2.2.215 tx (Series Text)

This element specifies text for a series name, without rich text formatting.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SerTx) is located in §A.5.1. *end note*]

21.2.2.216 txPr (Text Properties)

This element specifies text formatting. The lstStyle element is not supported.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.2.2.217 upBars (Up Bars)

This element specifies the up bars on the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBar) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.2.218 upDownBars (Up/Down Bars)

This element specifies the up and down bars.

[*Note*: The W3C XML Schema definition of this element's content model (CT_UpDownBars) is located in §A.5.1. *end note*]

21.2.2.219 userInterface (User Interface)

This element specifies that the protection applies to the user interface only, and not to changes made through the object model.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.220 userShapes (User Shapes)

This element shall specify the shapes drawn on top of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Drawing) is located in §A.5.2. *end note*]

21.2.2.221 userShapes (Reference to Chart Drawing Part)

This element specifies a relationship to a separate part which contains a drawing to be drawn on top of the chart.

Attributes	Description
id (Relationship Reference) Namespace: ttp://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the relationship for this Chart or Chart Drawing part. The type of relationship needed is specified by the parent element. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelId) is located in §A.5.1. *end note*]

21.2.2.222 v (Numeric Value)

This element specifies a numeric value.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.223 v (Text Value)

This element specifies a text value for a category axis label or a series name.

The possible values for this element are defined by the ST_Xstring simple type (§22.9.2.19).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Xstring) is located in §A.6.9. *end note*]

21.2.2.224 val (Values)

This element specifies the data values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.2.225 val (Error Bar Value)

This element specifies a value which is used with the errBar element to determine the length of the error bars.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.226 valAx (Value Axis)

This element specifies a value axis.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_ValAx) is located in §A.5.1. *end note*]

21.2.2.227 varyColors (Vary Colors by Point)

This element specifies that each data marker in the series has a different color.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.228 view3D (View In 3D)

This element specifies the 3-D view of the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_View3D) is located in §A.5.1. *end note*]

21.2.2.229 w (Width)

This element specifies the width (if Width Mode is Factor) or right (if Width Mode is Edge) of the chart element as a fraction of the width of the chart.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.2.230 wireframe (Wireframe)

This element specifies the surface chart is drawn as a wireframe.

Attributes	Description
val (Boolean Value)	Specifies a boolean value for the property defined by the parent XML element. A value of `1 or true` specifies that the property is applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of `0 or false` specifies that the property is not applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Boolean) is located in §A.5.1. *end note*]

21.2.2.231 wMode (Width Mode)

This element specifies how to interpret the Width element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.232 x (Left)

This element specifies the x location (left) of the chart element as a fraction of the width of the chart. If Left Mode is Factor, then the position is relative to the default position for the chart element.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.233 xMode (Left Mode)

This element specifies how to interpret the Left element for this manual layout.

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.234 xVal (X Values)

This element specifies the x values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AxDataSource) is located in §A.5.1. *end note*]

21.2.2.235 y (Top)

This element specifies the top of the chart element as a fraction of the height of the chart. If Top Mode is Factor, then the position is relative to the default position for the chart element.

Attributes	Description
val (Floating Point Value)	Specifies that the contents of this attribute contain a floating point number. The contents of this number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Double) is located in §A.5.1. *end note*]

21.2.2.236 yMode (Top Mode)

This element specifies how to interpret the Top element for this manual layout.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Layout Mode Value)	Specifies the layout mode for the width. The possible values for this attribute are defined by the ST_LayoutMode simple type (§21.2.3.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutMode) is located in §A.5.1. *end note*]

21.2.2.237 yVal (Y Values)

This element specifies the y values which shall be used to define the location of data markers on a chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumDataSource) is located in §A.5.1. *end note*]

21.2.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – Charts.

21.2.3.1 ST_AxisUnit (Axis Unit)

This simple type specifies that its contents contain a positive floating point number.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisUnit) is located in §A.5.1. *end note*]

21.2.3.2 ST_AxPos (Axis Position)

This simple type specifies the possible positions for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the axis shall be displayed at the bottom of the plot area.
l (Left)	Specifies that the axis shall be displayed at the left of the plot area.
r (Right)	Specifies that the axis shall be displayed at the right of the plot area.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
t (Top)	Specifies that the axis shall be displayed at the top of the plot area.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxPos) is located in §A.5.1. *end note*]

21.2.3.3 ST_BarDir (Bar Direction)

This simple type specifies the possible directions for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar)	Specifies that the chart is a bar chart - the data markers are horizontal rectangles.
col (Column)	Specifies that the chart is a column chart - the data markers are vertical rectangles.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BarDir) is located in §A.5.1. *end note*]

21.2.3.4 ST_BarGrouping (Bar Grouping)

This simple type specifies the possible groupings for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
clustered (Clustered)	Specifies that the chart series are drawn next to each other along the category axis.
percentStacked (100% Stacked)	Specifies that the chart series are drawn next to each other along the value axis and scaled to total 100%.
stacked (Stacked)	Specifies that the chart series are drawn next to each other on the value axis.
standard (Standard)	Specifies that the chart series are drawn next to each other on the depth axis.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BarGrouping) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.2.3.5 ST_BubbleScale (Bubble Scale)

This simple type specifies that its contents contain a percentage between 0% and 300%.

This simple type is a union of the following types:

- ST_BubbleScalePercent simple type (§21.2.3.58).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BubbleScale) is located in §A.5.1. *end note*]

21.2.3.6 ST_BuiltInUnit (Built-In Unit)

This simple type specifies the built in display units for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
billions (Billions)	Specifies the values on the chart shall be divided by 1,000,000,000.
hundredMillions (Hundred Millions)	Specifies the values on the chart shall be divided by 100,000,000.
hundreds (Hundreds)	Specifies the values on the chart shall be divided by 100.
hundredThousands (Hundred Thousands)	Specifies the values on the chart shall be divided by 100,000.
millions (Millions)	Specifies the values on the chart shall be divided by 1,000,000.
tenMillions (Ten Millions)	Specifies the values on the chart shall be divided by 10,000,000.
tenThousands (Ten Thousands)	Specifies the values on the chart shall be divided by 10,000.
thousands (Thousands)	Specifies the values on the chart shall be divided by 1,000.
trillions (Trillions)	Specifies the values on the chart shall be divided by 1,000,000,000,000.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BuiltInUnit) is located in §A.5.1. *end note*]

21.2.3.7 ST_CrossBetween (Cross Between)

This simple type specifies the possible crossing states of an axis.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
between (Between)	Specifies the value axis shall cross the category axis between data markers.
midCat (Midpoint of Category)	Specifies the value axis shall cross the category axis at the midpoint of a category.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CrossBetween) is located in §A.5.1. *end note*]

21.2.3.8 ST_Crosses (Crosses)

This simple type specifies the possible crossing points for an axis.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
autoZero (Axis Crosses at Zero)	The category axis crosses at the zero point of the value axis (if possible), or the minimum value (if the minimum is greater than zero) or the maximum (if the maximum is less than zero).
max (Maximum)	The axis crosses at the maximum value
min (Minimum)	Axis crosses at the minimum value of the chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Crosses) is located in §A.5.1. *end note*]

21.2.3.9 ST_DepthPercent (Depth Percent)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

This simple type is a union of the following types:

- ST_DepthPercentWithSymbol simple type (§21.2.3.51).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DepthPercent) is located in §A.5.1. *end note*]

21.2.3.10 ST_DispBlanksAs (Display Blanks As)

This simple type specifies the possible ways to display blanks.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
gap (Gap)	Specifies that blank values shall be left as a gap.
span (Span)	Specifies that blank values shall be spanned with a line.
zero (Zero)	Specifies that blank values shall be treated as zero.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DispBlanksAs) is located in §A.5.1. *end note*]

21.2.3.11 ST_DLblPos (Data Label Position)

This simple type specifies the possible positions for a data label.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that data labels shall be displayed below the data marker.
bestFit (Best Fit)	Specifies that data labels shall be displayed in the best position.
ctr (Center)	Specifies that data labels shall be displayed centered on the data marker.
inBase (Inside Base)	Specifies that data labels shall be displayed inside the base of the data marker.
inEnd (Inside End)	Specifies that data labels shall be displayed inside the end of the data marker.
l (Left)	Specifies that data labels shall be displayed to the left of the data marker.
outEnd (Outside End)	Specifies that data labels shall be displayed outside the end of the data marker.
r (Right)	Specifies that data labels shall be displayed to the right of the data marker.
t (Top)	Specifies that data labels shall be displayed above the data marker.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DLblPos) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.3.12 ST_ErrBarType (Error Bar Type)

This simple type specifies the possible ways to draw an error bar.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
both (Both)	Specifies that error bars shall be shown in the positive and negative directions.
minus (Minus)	Specifies that error bars shall be shown in the negative direction only.
plus (Plus)	Specifies that error bars shall be shown in the positive direction only.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrBarType) is located in §A.5.1. *end note*]

21.2.3.13 ST_ErrDir (Error Bar Direction)

This simple type specifies the possible directions for error bars.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
x (X)	Specifies that error bars shall be shown in the x direction.
y (Y)	Specifies that error bars shall be shown in the y direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrDir) is located in §A.5.1. *end note*]

21.2.3.14 ST_ErrValType (Error Value Type)

This simple type specifies the possible ways to determine the length of the error bars

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cust (Custom Error Bars)	Specifies that the length of the error bars shall be determined by the Plus and Minus elements.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
fixedVal (Fixed Value)	Specifies that the length of the error bars shall be the fixed value determined by Error Bar Value.
percentage (Percentage)	Specifies that the length of the error bars shall be Error Bar Value percent of the data.
stdDev (Standard Deviation)	Specifies that the length of the error bars shall be Error Bar Value standard deviations of the data.
stdErr (Standard Error)	Specifies that the length of the error bars shall be Error Bar Value standard errors of the data.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ErrValType) is located in §A.5.1. *end note*]

21.2.3.15 ST_FirstSliceAng (First Slice Angle)

This simple type specifies that its contents contain an integer between 0 and 360.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 360.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FirstSliceAng) is located in §A.5.1. *end note*]

21.2.3.16 ST_GapAmount (GapAmount)

This simple type specifies that its contents contain a percentage between 0% and 500%.

This simple type is a union of the following types:

- ST_GapAmountPercent simple type (§21.2.3.53).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GapAmount) is located in §A.5.1. *end note*]

21.2.3.17 ST_Grouping (Grouping)

This simple type specifies the possible groupings for a bar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
percentStacked (100% Stacked)	Specifies that the chart series are drawn next to each other along the value axis and scaled to total 100%.
stacked (Stacked)	Specifies that the chart series are drawn next to each other on the value axis.
standard (Standard)	Specifies that the chart series are drawn on the value axis.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Grouping) is located in §A.5.1. *end note*]

21.2.3.18 ST_HoleSize (Hole Size)

This simple type specifies that its contents contain a percentage between 1% and 90%.

This simple type is a union of the following types:

- ST_HoleSizePercent simple type (§21.2.3.55).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HoleSize) is located in §A.5.1. *end note*]

21.2.3.19 ST_HPercent (Height Percent)

This simple type specifies that its contents contain a percentage between 5% and 500%.

This simple type is a union of the following types:

- ST_HPercentWithSymbol simple type (§21.2.3.52).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HPercent) is located in §A.5.1. *end note*]

21.2.3.20 ST_LayoutMode (Layout Mode)

This simple type specifies the possible ways to store a chart element's position.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
edge (Edge)	Specifies that the Width or Height shall be interpreted as the Right or Bottom of the chart element.
factor (Factor)	Specifies that the Width or Height shall be interpreted as the Width or Height of the chart element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutMode) is located in §A.5.1. *end note*]

21.2.3.21 ST_LayoutTarget (Layout Target)

This simple type specifies the possible ways to layout the plot area.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
inner (Inner)	Specifies that the plot area size shall determine the size of the plot area, not including the tick marks and axis labels.
outer (Outer)	Specifies that the plot area size shall determine the size of the plot area, the tick marks, and the axis labels.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutTarget) is located in §A.5.1. *end note*]

21.2.3.22 ST_LblAlgn (Label Alignment)

This simple type specifies the possible ways to align the tick labels.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies that the text shall be centered.
l (Left)	Specifies that the text shall be left justified.
r (Right)	Specifies that the text shall be right justified.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LblAlgn) is located in §A.5.1. *end note*]

21.2.3.23 ST_LblOffset (Label Offset)

This simple type specifies that its contents contain a percentage of the default value, between 0% and 1000%.

This simple type is a union of the following types:

- ST_LblOffsetPercent simple type (§21.2.3.56).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LblOffset) is located in §A.5.1. *end note*]

21.2.3.24 ST_LegendPos (Legend Position)

This simple type specifies the possible positions for a legend.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the legend shall be drawn at the bottom of the chart.
l (Left)	Specifies that the legend shall be drawn at the left of the chart.
r (Right)	Specifies that the legend shall be drawn at the right of the chart.
t (Top)	Specifies that the legend shall be drawn at the top of the chart.
tr (Top Right)	Specifies that the legend shall be drawn at the top right of the chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LegendPos) is located in §A.5.1. *end note*]

21.2.3.25 ST_LogBase (Logarithmic Base)

This simple type specifies that its contents contain a floating point number greater than or equal to two.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a maximum value of less than or equal to 1000.
- This simple type has a minimum value of greater than or equal to 2.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LogBase) is located in §A.5.1. *end note*]

21.2.3.26 ST_MarkerSize (Marker Size)

This simple type specifies that its contents contain an integer between 2 and 72, whose contents are a size in points.

This simple type's contents are a restriction of the W3C XML Schema unsignedByte datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 2.
- This simple type has a maximum value of less than or equal to 72.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerSize) is located in §A.5.1. *end note*]

21.2.3.27 ST_MarkerStyle (Marker Style)

This picture shows each of the marker styles. Black is used as the line color, while red is used as the fill color. The height of the dash and the dot are 1/5th of the height of the marker. The width of the dot is 1/2 the width of the marker. The dash and dot have fills as well, but the markers need to be made quite large before these are visible.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
circle (Circle)	Specifies a circle shall be drawn at each data point.
dash (Dash)	Specifies a dash shall be drawn at each data point.
diamond (Diamond)	Specifies a diamond shall be drawn at each data point.
dot (Dot)	Specifies a dot shall be drawn at each data point.
none (None)	Specifies nothing shall be drawn at each data point.
picture (Picture)	Specifies a picture shall be drawn at each data point.
plus (Plus)	Specifies a plus shall be drawn at each data point.
square (Square)	Specifies a square shall be drawn at each data point.
star (Star)	Specifies a star shall be drawn at each data point.
triangle (Triangle)	Specifies a triangle shall be drawn at each data point.
x (X)	Specifies an X shall be drawn at each data point.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerStyle) is located in §A.5.1. *end note*]

21.2.3.28 ST_OfPieType (Pie of Pie or Bar of Pie Type)

This simple type specifies the possible modes of Pie of Pie or Bar of Pie charts.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar)	Specifies that the chart is a bar of pie chart, not a pie of pie chart.
pie (Pie)	Specifies that the chart is pie of pie chart, not a bar of pie chart.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OfPieType) is located in §A.5.1. *end note*]

21.2.3.29 ST_Order (Order)

This simple type specifies that its contents contain an integer between 2 and 6, whose contents are the order of the trendline polynomial.

This simple type's contents are a restriction of the W3C XML Schema unsignedByte datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 2.
- This simple type has a maximum value of less than or equal to 6.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Order) is located in §A.5.1. *end note*]

21.2.3.30 ST_Orientation (Orientation)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
maxMin (Maximum to Minimum)	Specifies that the values on the axis shall be reversed so they go from maximum to minimum.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
minMax (Minimum to Maximum)	Specifies that the axis values shall be in the usual order, minimum to maximum.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Orientation) is located in §A.5.1. *end note*]

21.2.3.31 ST_Overlap (Overlap)

This simple type specifies that its contents contain a percentage between -100% and 100%.

This simple type is a union of the following types:

- ST_OverlapPercent simple type (§21.2.3.57).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Overlap) is located in §A.5.1. *end note*]

21.2.3.32 ST_PageSetupOrientation (Printed Page Orientation)

This simple type specifies the page orientation of the printed page(s) on which this chart shall appear.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
default (Default Page Orientation)	Specifies that the page orientation shall be the default orientation of the system.
landscape (Landscape Page)	Specifies that the printed page shall have landscape orientation.
portrait (Portrait Page)	Specifies that the printed page shall have portrait orientation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PageSetupOrientation) is located in §A.5.1. *end note*]

21.2.3.33 ST_Period (Period)

This simple type specifies that its contents contain an integer greater than or equal to 2.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 2.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
auto (Auto)	Specifies an application-specific marker shall be drawn at each data point.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Period) is located in §A.5.1. *end note*]

21.2.3.34 ST_Perspective (Perspective)

This simple type specifies that its contents contain an integer between 0 and 100, whose contents are a percentage.

This simple type's contents are a restriction of the W3C XML Schema unsignedByte datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 240.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Perspective) is located in §A.5.1. *end note*]

21.2.3.35 ST_PictureFormat (Picture Format)

This simple type specifies the possible ways to place a picture on a data point, series, wall, or floor.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
stack (Stack)	Specifies that the picture shall be stacked.
stackScale (Stack and Scale)	Specifies that the picture shall be stacked after being scaled so that it's height is one Picture Stack Unit. Does not apply to walls or floor.
stretch (Stretch)	Specifies that the picture shall be anisotropic stretched to fill the data point, series, wall or floor.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PictureFormat) is located in §A.5.1. *end note*]

21.2.3.36 ST_PictureStackUnit (Picture Stack Unit)

This simple type specifies that its contents contain a floating point number greater than zero.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than 0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PictureStackUnit) is located in §A.5.1. *end note*]

21.2.3.37 ST_RadarStyle (Radar Style)

This simple type specifies the possible styles of radar chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
filled (Filled)	Specifies that the radar chart shall be filled and have lines but no markers.
marker (Marker)	Specifies that the radar chart shall have lines and markers but no fill.
standard (Standard)	Specifies that the radar chart shall have lines but no markers and no fill.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RadarStyle) is located in §A.5.1. *end note*]

21.2.3.38 ST_RotX (X Rotation)

This simple type specifies that its contents contain an integer between -90 and 90, whose contents are an angle in degrees.

This simple type's contents are a restriction of the W3C XML Schema byte datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -90.
- This simple type has a maximum value of less than or equal to 90.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotX) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.3.39 ST_RotY (Y Rotation)

This simple type specifies that its contents contain an integer between 0 and 360, whose contents are an angle in degrees.

This simple type's contents are a restriction of the W3C XML Schema unsignedShort datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 360.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotY) is located in §A.5.1. *end note*]

21.2.3.40 ST_ScatterStyle (Scatter Style)

This simple type specifies the possible styles of scatter chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
line (Line)	Specifies the points on the scatter chart shall be connected with straight lines but markers shall not be drawn.
lineMarker (Line with Markers)	Specifies the points on the scatter chart shall be connected with straight lines and markers shall be drawn.
marker (Marker)	Specifies the points on the scatter chart shall not be connected with lines and markers shall be drawn.
none (None)	Specifies the points on the scatter chart shall not be connected with straight lines and markers shall not be drawn.
smooth (Smooth)	Specifies the the points on the scatter chart shall be connected with smoothed lines and markers shall not be drawn.
smoothMarker (Smooth with Markers)	Specifies the the points on the scatter chart shall be connected with smoothed lines and markers shall be drawn.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ScatterStyle) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.2.3.41 ST_SecondPieSize (Second Pie Size)

This simple type specifies that its contents contain a percentage between 5% and 200%, whose contents consist of a percentage.

This simple type is a union of the following types:

- ST_SecondPieSizePercent simple type (§21.2.3.55).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondPieSize) is located in §A.5.1. *end note*]

21.2.3.42 ST_Shape (Shape)

This simple type specifies the possible shapes for a 3-D data marker.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
box (Box)	Specifies the chart shall be drawn with a box shape.
cone (Cone)	Specifies the chart shall be drawn as a cone, with the base of the cone on the floor and the point of the cone at the top of the data marker.
coneToMax (Cone to Max)	Specifies the chart shall be drawn with truncated cones such that the point of the cone would be the maximum data value.
cylinder (Cylinder)	Specifies the chart shall be drawn as a cylinder.
pyramid (Pyramid)	Specifies the chart shall be drawn as a rectangular pyramid, with the base of the pyramid on the floor and the point of the pyramid at the top of the data marker.
pyramidToMax (Pyramid to Maximum)	Specifies the chart shall be drawn with truncated cones such that the point of the cone would be the maximum data value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Shape) is located in §A.5.1. *end note*]

21.2.3.43 ST_SizeRepresents (Size Represents)

This simple type specifies the possible ways to represent data as bubble chart sizes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
area (Bubble Size Represents Area)	Specifies the area of the bubbles shall be proportional to the bubble size value.
w (Bubble Size Represents Width)	Specifies the radius of the bubbles shall be proportional to the bubble size value.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SizeRepresents) is located in §A.5.1. *end note*]

21.2.3.44 ST_Skip (Skip)

This simple type specifies that its contents contain an integer greater than or equal to one.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Skip) is located in §A.5.1. *end note*]

21.2.3.45 ST_SplitType (Split Type)

This simple type specifies the possible ways to split a pie of pie or bar of pie chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Default Split)	Specifies the data points shall be split using the default mechanism for this chart type.
cust (Custom Split)	Specifies the data points shall be split between the pie and the second chart according to the Custom Split values.
percent (Split by Percentage)	Specifies the data points shall be split between the pie and the second chart by putting the points with percentage less than Split Position percent in the second chart.
pos (Split by Position)	Specifies the data points shall be split between the pie and the second chart by putting the last Split Position of the data points in the second chart
val (Split by Value)	Specifies the data points shall be split between the pie and the second chart by putting the data points with

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	value less than Split Position in the second chart

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SplitType) is located in §A.5.1. *end note*]

21.2.3.46 ST_Style (Style)

This simple type specifies that its contents contain an integer between 1 and 48. The value determines the default formatting for all chart elements through the tables described below.

The default font is the minor font as defined by the document's theme. The default font size for each element is the font size of the chart, except for the title which is always 120% the font size of the chart. If the chart does not have a font size set, then the default font size is 10. Axis titles and chart titles are bold by default, while all other chart elements are normal. The default font color is the same as the Axis & Major Gridlines Line Color.

The default line style, fill style, and effect style are determined by the tables below. Each of the default includes a themed line, fill, or effect (None, Subtle, Moderate, or Intense) and a color to be used when applying that line, fill, or effect. In some cases, both the themed formatting and the color vary per style, in other cases they do not. The default line width is determined by the theme, except for lines for data points it is multipled by the line width value given in the table.

Table 1: Chart element defaults

This table lists whether the default is constant or whether it depends on the ST_Style using one of the later tables.

	Line		Fill		Effect	
Chart Element	Themed Line	Color	Themed Fill	Color	Themed Effect	Color
Axis	Subtle	Table 2	No Fill		No Effect	
Axis Title	No Line		No Fill		No Effect	
Chart Area	Table 2		Subtle	Table 3	No Effect	
Chart Title	No Line		No Fill		No Effect	
Data Labels	No Line		No Fill		No Effect	
Data Table	Subtle	Table 2	No Fill		No Effect	
Down Bars	Table 4		Table 4		Table 4	dk1
Fills for Data Points (2-D)	Subtle	Table 5	Table 5		Table 5	dk1
Fills for Data Points (3-D)	Subtle	Table 5	Table 5		Table 5	dk1
Floor	Table 2		Table 3		No Effect	
Legend	No Line		No Fill		No Effect	
Lines for Data Points	Subtle	Table 5	N/A		No Effect	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	Line		Fill		Effect	
Major Gridlines	Subtle	Table 2	N/A		No Effect	
Markers for Data Points	Subtle	Table 5	Table 5		Table 5	dk1
Minor Gridlines	Subtle	Table 2	N/A		No Effect	
Other Lines	Subtle	Table 2	N/A		No Effect	
Plot Area (2-D charts)	No Line		Subtle	Table 3	No Effect	
Plot Area (3-D charts)	No Line		No Fill		No Effect	
Trendline Labels	No Line		No Fill		No Effect	
Up Bars	Table 4		Table 4		Table 4	dk1
Walls	No Line		Table 3		No Effect	

Other Lines includes Drop Lines, Error Bars, High Low Lines, Leader Lines, Series Lines, and Trendlines.

Fills for Data Points (2-D) includes 2-D bar, filled radar, stock, bubble, pie, doughnut and area charts.

Fills for Data Points (3-D) includes all 3-D charts.

Lines for Data Points includes lines on 2-D line, scatter, bubble, and radar charts.

Markers for Data Points includes markers on 2-D line, scatter, and radar charts.

Table 2: Default line formatting per chart style

This table lists line formatting for several chart elements by style.

Style	Axis & Major Gridlines	Minor Gridlines	Chart Area, Data Table, & Floor	Other Lines	Floor & Chart Area
	Color	Color	Color	Color	Themed Line
1-32	tx1	50% tint of tx1	75% tint of tx1	tx1	Subtle
33-34	dk1	50% tint of tx1	75% tint of dk1	dk1	Subtle
35-40	dk1	50% tint of tx1	75% tint of dk1	25% shade of dk1	Subtle
41-48	dk1	90% tint of tx1	lt1	lt1	No Line

Table 3: Default fill formatting per chart style

This table lists fill formatting for several chart elements by style.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Style	Chart Area	Floor, Walls & Plot Area (2-D)	Floor & Walls
	Color	Color	Themed Fill
1-32	bg1	bg1	No Fill
33-34	lt1	20% tint of dk1	Subtle
35-40	lt1	accent1-6	Subtle
41-48	dk1	95% tint of dk1	Subtle

Table 4: Up and down bars default formatting per chart style

This table lists line, fill, and effect formatting for up and down bars by style. The color listed as accent1-6 means that the first style uses accent1, the next uses accent2, up to the sixth uses accent6.

	Up Bars	Down Bars	Up & Down Bars			
			Themed			
Style	Fill Color	Fill Color	Fill	Effect	Line	Line Color
1	25% tint of dk1	85% tint of dk1	Subtle	None	Subtle	tx1
2	5% tint of dk1	95% tint of dk1	Subtle	None	Subtle	tx1
3-8	25% tint of accent1-6	25% shade of accent1-6	Subtle	None	Subtle	tx1
9	25% tint of dk1	85% tint of dk1	Subtle	Subtle	Subtle	tx1
10	5% tint of dk1	95% tint of dk1	Subtle	Subtle	Subtle	tx1
11-16	25% tint of accent1-6	25% shade of accent1-6	Subtle	Subtle	Subtle	tx1
17	25% tint of dk1	85% tint of dk1	Intense	Moderate	No ne	None
18	5% tint of dk1	95% tint of dk1	Intense	Moderate	None	None
19-24	25% tint of accent1-6	25% shade of accent1-6	Intense	Moderate	None	None
25	25% tint of dk1	85% tint of dk1	Intense	Intense	None	None
26	5% tint of dk1	95% tint of dk1	Intense	Intense	None	None

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

	Up Bars	Down Bars	Up & Down Bars			
27-32	25% tint of accent1-6	25% shade of accent1-6	Intense	Intense	None	None
33	lt1	85% tint of dk1	Subtle	None	Subtle	dk1
34	lt1	95% tint of dk1	Subtle	None	Subtle	dk1
35-40	lt1	25% shade of accent1-6	Subtle	None	Subtle	25% shade of accent1-6
41	25% tint of dk1	85% tint of dk1	Intense	Intense	None	None
42	lt1	dk1	Intense	Intense	None	None
43-48	25% tint of accent1-6	25% shade of accent1-6	Intense	Intense	None	None

Table 5: Default data point formatting per chart style

This table lists line, fill, and effect formatting for data points by style. Some of the formatting is a repeating pattern described in the next table (denoted by "Pattern"). Other use a fade pattern in which the first series is a certain% shade of the color listed and the last series is an certain% tint of the color listed. The intermediate colors are linearly interpolated by shade and tint to the listed color. The starting shade% and ending tint% are determined by the spreadsheet application, where shades are always darker than tints.

[*Note*: A suggested way to implement this percentage is by using the formula: shade/tint percent = -70 + 140 * (SeriesFormattingIndex / (HighestFormattingIndexOfAllSeriesOnAllGraphs + 1)). In this case the series formatting index is the `idx` attribute value for the series. Negative outputs are shades, and positive outputs are tints. *end note*]

[*Note*: The arrows across the top of the table indicate which columns apply for that row: i.e., for Fills, Lines, or Markers. So, for example, only the last two columns for Lines for Data Points apply. *end note*]

Fills for Data Points (2-D)	↓	↓		↓	↓			
Fills for Data Points (3-D)		↓	↓	↓	↓			
Lines for Data Points						↓	↓	
Markers for Data Points	↓	↓					↓	

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Style	Effect Themed Effect	Fill Themed Fill	Fill Pattern	Fill Themed Fill	Line Themed Line	Line Color or Pattern	Line Width	Line Pattern
1	No Effect	Subtle	1	Subtle	No Line		3	1
2	No Effect	Subtle	2	Subtle	No Line		3	2
3	No Effect	Subtle	accent1 fade	Subtle	No Line		3	accent1 fade
4	No Effect	Subtle	accent2 fade	Subtle	No Line		3	accent2 fade
5	No Effect	Subtle	accent3 fade	Subtle	No Line		3	accent3 fade
6	No Effect	Subtle	accent4 fade	Subtle	No Line		3	accent4 fade
7	No Effect	Subtle	accent5 fade	Subtle	No Line		3	accent5 fade
8	No Effect	Subtle	accent6 fade	Subtle	No Line		3	accent6 fade
9	Subtle	Subtle	1	Subtle	Subtle	lt1	5	1
10	Subtle	Subtle	2	Subtle	Subtle	lt1	5	2
11	Subtle	Subtle	accent1 fade	Subtle	Subtle	lt1	5	accent1 fade
12	Subtle	Subtle	accent2 fade	Subtle	Subtle	lt1	5	accent2 fade
13	Subtle	Subtle	accent3 fade	Subtle	Subtle	lt1	5	accent3 fade
14	Subtle	Subtle	accent4 fade	Subtle	Subtle	lt1	5	accent4 fade
15	Subtle	Subtle	accent5 fade	Subtle	Subtle	lt1	5	accent5 fade
16	Subtle	Subtle	accent6	Subtle	Subtle	lt1	5	accent6

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

			fade				fade
17	Moderate	Intense	1	Subtle	No Line	5	1
18	Moderate	Intense	2	Intense	No Line	5	2
19	Moderate	Intense	accent1 fade	Subtle	No Line	5	accent1 fade
20	Moderate	Intense	accent2 fade	Subtle	No Line	5	accent2 fade
21	Moderate	Intense	accent3 fade	Subtle	No Line	5	accent3 fade
22	Moderate	Intense	accent4 fade	Subtle	No Line	5	accent4 fade
23	Moderate	Intense	accent5 fade	Subtle	No Line	5	accent5 fade
24	Moderate	Intense	accent6 fade	Subtle	No Line	5	accent6 fade
25	Intense	Intense	1	Subtle	No Line	7	1
26	Intense	Intense	2	Intense	No Line	7	2
27	Intense	Intense	accent1 fade	Subtle	No Line	7	accent1 fade
28	Intense	Intense	accent2 fade	Subtle	No Line	7	accent2 fade
29	Intense	Intense	accent3 fade	Subtle	No Line	7	accent3 fade
30	Intense	Intense	accent4 fade	Subtle	No Line	7	accent4 fade
31	Intense	Intense	accent5 fade	Subtle	No Line	7	accent5 fade
32	Intense	Intense	accent6 fade	Subtle	No Line	7	accent6 fade
33	No Effect	Subtle	1	Subtle	Subtle	5	1
					50% shade		

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

						of dk1		
34	No Effect	Subtle	2	Subtle	Subtle	Pattern 3	5	2
35	No Effect	Subtle	accent1 fade	Subtle	Subtle	50% shade of accent1	5	accent1 fade
36	No Effect	Subtle	accent2 fade	Subtle	Subtle	50% shade of accent2	5	accent2 fade
37	No Effect	Subtle	accent3 fade	Subtle	Subtle	50% shade of accent3	5	accent3 fade
38	No Effect	Subtle	accent4 fade	Subtle	Subtle	50% shade of accent4	5	accent4 fade
39	No Effect	Subtle	accent5 fade	Subtle	Subtle	50% shade of accent5	5	accent5 fade
40	No Effect	Subtle	accent6 fade	Subtle	Subtle	50% shade of accent6	5	accent6 fade
41	Intense	Intense	4	Subtle	No Line		5	4
42	Intense	Intense	2	Intense	No Line		5	2
43	Intense	Intense	accent1 fade	Subtle	No Line		5	accent1 fade
44	Intense	Intense	accent2 fade	Subtle	No Line		5	accent2 fade
45	Intense	Intense	accent3 fade	Subtle	No Line		5	accent3 fade
46	Intense	Intense	accent4 fade	Subtle	No Line		5	accent4 fade
47	Intense	Intense	accent5 fade	Subtle	No Line		5	accent5 fade
48	Intense	Intense	accent6 fade	Subtle	No Line		5	accent6 fade

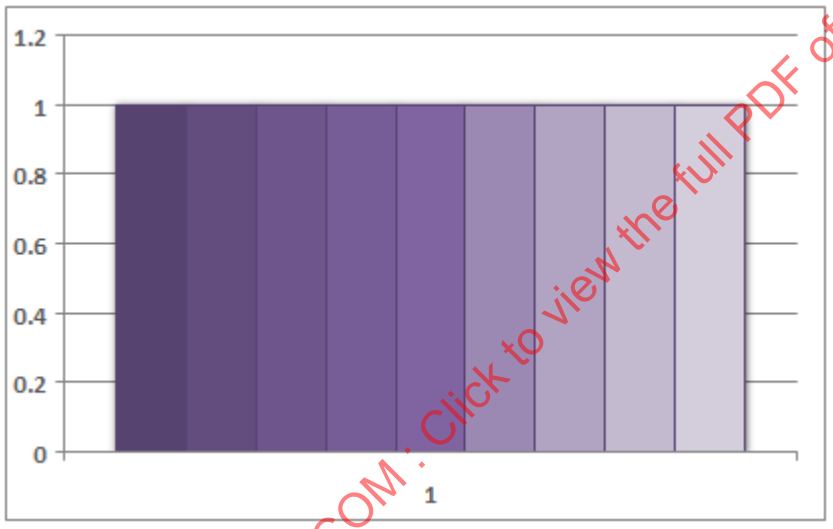
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Table 6: Default data point formatting per data point

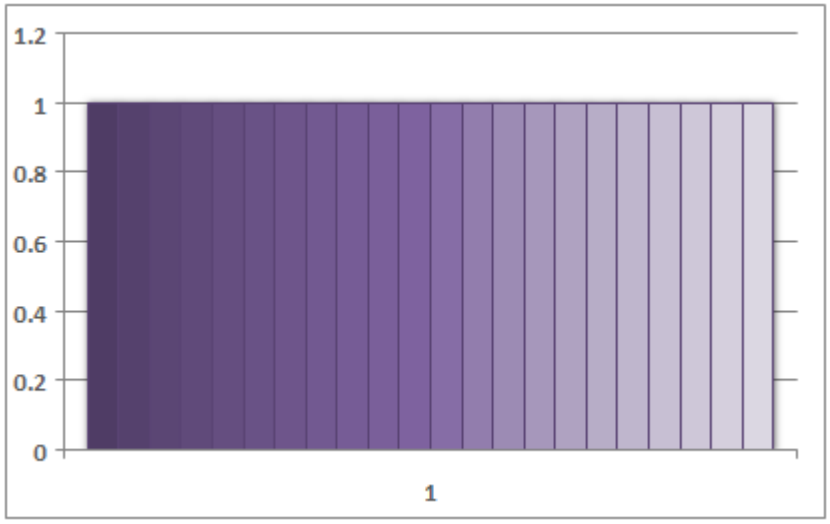
This table contains the formatting patterns used for each data point when there are is one series, or each series when there is just one series. The patterns in this table can repeat, see the pictures below the table for an illustration.

Pattern	Data Point 1	Data Point 2	Data Point 3	Data Point 4	Data Point 5	Data Point 6
1	88.5% tint of dk1	55% tint of dk1	78% tint of dk1	92.5% tint of dk1	70% tint of dk1	30% tint of dk1
2	accent1	accent2	accent3	accent4	Accent5	accent6
3	50% shade of accent1	50% shade of accent2	50% shade of accent3	50% shade of accent4	50% shade of accent5	50% shade of accent6
4	5% tint of dk1	55% tint of dk1	78% tint of dk1	15% tint of dk1	70% tint of dk1	30% tint of dk1

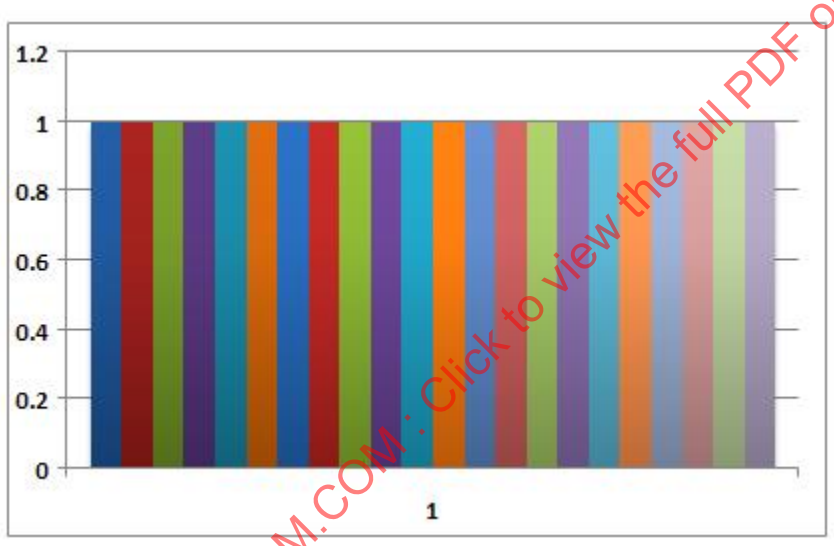


IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved



Above are two charts showing the same monochromatic pattern, just with a different number of series. [*Note*: These charts just have tint and shade adjusted for each series beginning and ending at the same values. *end note*]



Above is a pattern that has 6 distinct accent colors. The colors repeat every 6 data points, but the tint/shade is changed for each set.

This simple type's contents are a restriction of the W3C XML Schema unsignedByte datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 48.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Style) is located in §A.5.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.2.3.47 ST_TickLblPos (Tick Label Position)

This simple type specifies the possible positions for tick labels.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
high (High)	Specifies the axis labels shall be at the high end of the perpendicular axis.
low (Low)	Specifies the axis labels shall be at the low end of the perpendicular axis.
nextTo (Next To)	Specifies the axis labels shall be next to the axis.
none (None)	Specifies the axis labels are not drawn.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TickLblPos) is located in §A.5.1. *end note*]

21.2.3.48 ST_TickMark (Tick Mark)

This simple type specifies the possible positions for tick marks.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cross (Cross)	Specifies the tick marks shall cross the axis.
in (Inside)	Specifies the tick marks shall be inside the plot area.
none (None)	Specifies there shall be no tick marks.
out (Outside)	Specifies the tick marks shall be outside the plot area.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TickMark) is located in §A.5.1. *end note*]

21.2.3.49 ST_TimeUnit (Time Unit)

This simple type specifies a unit of time.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
days (Days)	Specifies the chart data shall be shown in days.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
months (Months)	Specifies the chart data shall be shown in months.
years (Years)	Specifies the chart data shall be shown in years.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TimeUnit) is located in §A.5.1. *end note*]

21.2.3.50 ST_TrendlineType (Trendline Type)

This simple type specifies all styles of trendline which are available for series in a chart.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
exp (Exponential)	Specifies the trendline shall be an exponential curve in the form $y = ab^x$.
linear (Linear)	Specifies the trendline shall be a line in the form $y = mx + b$.
log (Logarithmic)	Specifies the trendline shall be a logarithmic curve in the form $y = a \log x + b$, where log is the natural logarithm.
movingAvg (Moving Average)	Specifies the trendline shall be a moving average of period Period.
poly (Polynomial)	Specifies the trendline shall be a polynomial curve of order Order in the form $y = ax^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g$.
power (Power)	Specifies the trendline shall be a power curve in the form $y = ax^b$.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TrendlineType) is located in §A.5.1. *end note*]

21.2.3.51 ST_DepthPercentWithSymbol (Depth Percent with Symbol)

This simple type specifies that its contents contain a percentage between 20% and 2000%.

The simple type's contents shall match the following regular expression pattern:
0*(([2-9][0-9])|([1-9][0-9][0-9])|(1[0-9][0-9][0-9])|2000)%.

21.2.3.52 ST_HPercentWithSymbol (Height Percent with Symbol)

This simple type specifies that its contents contain a percentage between 5% and 500%.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The simple type's contents shall match the following regular expression pattern:

0*(([5-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%.

21.2.3.53 ST_GapAmountPercent (Gap Amount Percentage)

This simple type specifies that its contents contain a percentage between 0% and 500%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-4][0-9][0-9])|500)%.

21.2.3.54 ST_SecondPieSizePercent (Second Pie Size Percentage)

This simple type specifies that its contents contain a percentage between 5% and 200%.

The simple type's contents shall match the following regular expression pattern:

0* (([5-9])|([1-9][0-9])|(1[0-9][0-9])|200)%.

21.2.3.55 ST_HoleSizePercent (Hole Size Percentage)

This simple type specifies that its contents contain a percentage between 1% and 90%.

The simple type's contents shall match the following regular expression pattern: 0*([1-9]|([1-8][0-9])|90)%.

21.2.3.56 ST_LblOffsetPercent (Label Offset Percentage)

This simple type specifies that its contents contain a percentage between 0% and 1000%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-9][0-9][0-9])|1000)%.

21.2.3.57 ST_OverlapPercent (Overlap Percentage)

This simple type specifies that its contents contain a percentage between -100% and 100%.

The simple type's contents shall match the following regular expression pattern:

(-?0*(([0-9])|([1-9][0-9])|100))%.

21.2.3.58 ST_BubbleScalePercent (Bubble Scale Percentage)

This simple type specifies that its contents contain a percentage between 0% and 300%.

The simple type's contents shall match the following regular expression pattern:

0*(([0-9])|([1-9][0-9])|([1-2][0-9][0-9])|300)%.

21.2.3.59 ST_Thickness (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is a union of the following types:

- ST_ThicknessPercent simple type (§21.2.3.60).

21.2.3.60 ST_ThicknessPercent (Thickness Percentage)

This simple type specifies that its contents contain a percentage.

The simple type's contents shall match the following regular expression pattern: ([0-9]+)%.

21.3 DrawingML - Chart Drawings

Within a chart, it is sometimes necessary to include DrawingML elements (shapes or pictures) which should be a child object within the parent chart. This relationship allows those elements to optionally be resized with the chart, automatically moved with the chart, etc.

The Chart Drawing namespace acts in this capacity, specifying all information necessary to anchor and display DrawingML objects within a chart.

21.3.1 Table of Contents

This subclause is informative.

21.3.2 Elements 3470
21.3.2.1 absSizeAnchor (Absolute Anchor Shape Size) 3470
21.3.2.2 blipFill (Picture Fill) 3470
21.3.2.3 cNvCxnSpPr (Non-Visual Connection Shape Drawing Properties) 3473
21.3.2.4 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties) 3473
21.3.2.5 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties) 3473
21.3.2.6 cNvPicPr (Non-Visual Picture Drawing Properties) 3473
21.3.2.7 cNvPr (Non-Visual Drawing Properties) 3474
21.3.2.8 cNvSpPr (Non-Visual Shape Drawing Properties) 3476
21.3.2.9 cxnSp (Connection Shape) 3477
21.3.2.10 ext (Shape Extent) 3479
21.3.2.11 from (Starting Anchor Point) 3479
21.3.2.12 graphicFrame (Graphic Frame) 3480
21.3.2.13 grpSp (Group Shape) 3481
21.3.2.14 grpSpPr (Group Shape Properties) 3482
21.3.2.15 nvCxnSpPr (Connector Non Visual Properties) 3482
21.3.2.16 nvGraphicFramePr (Non-Visual Graphic Frame Properties) 3483
21.3.2.17 nvGrpSpPr (Non-Visual Group Shape Properties) 3483
21.3.2.18 nvPicPr (Non-Visual Picture Properties) 3483
21.3.2.19 nvSpPr (Non-Visual Shape Properties) 3483
21.3.2.20 pic (Picture) 3484
21.3.2.21 relSizeAnchor (Relative Anchor Shape Size) 3485
21.3.2.22 sp (Shape) 3485
21.3.2.23 spPr (Shape Properties) 3487

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.3.2.24 style (Shape Style) 3487
21.3.2.25 to (Ending Anchor Point) 3488
21.3.2.26 txBody (Shape Text Body) 3488
21.3.2.27 x (Relative X Coordinate) 3488
21.3.2.28 xfrm (Graphic Frame Transform) 3489
21.3.2.29 y (Relative Y Coordinate) 3490

21.3.3 Simple Types 3490

21.3.3.1 ST_MarkerCoordinate (Chart Marker Coordinate Value) 3490

End of informative text.

21.3.2 Elements

The following element define the contents of the ChartDrawing namespace:

21.3.2.1 absSizeAnchor (Absolute Anchor Shape Size)

This element specifies that the shape described here to reside within a chart should be sized based on relative anchor points. This is achieved via two elements. The from element specifies the top left corner of the shape bounding box in a RTL(right-to-left) implementation. The ext element then specifies the bottom right corner of the shape bounding box in a RTL(right-to-left) implementation and thus the size of the shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_AbsSizeAnchor) is located in §A.5.2. *end note*]

21.3.2.2 blipFill (Picture Fill)

This element specifies the kind of picture fill that the picture object has. Because a picture has a picture fill already by default, it is possible to have two fills specified for a picture object. An example of this is shown below.

[*Example*: Consider the picture below that has a blip fill applied to it. The image used to fill this picture object has transparent pixels instead of white pixels.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  …
</pic:pic>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The above picture object is shown as an example of this fill kind. *end example*]

[*Example*: Consider now the same picture object but with an additional gradient fill applied within the shape properties portion of the picture.

```
<pic:pic>
  …
  <pic:blipFill>
    <a:blip r:embed="rId2"/>
    <a:stretch>
      <a:fillRect/>
    </a:stretch>
  </pic:blipFill>
  <pic:spPr>
    <a:gradFill>
      <a:gsLst>
        <a:gs pos="0">
          <a:schemeClr val="tx2">
            <a:shade val="50000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="39999">
          <a:schemeClr val="tx2">
            <a:tint val="20000"/>
          </a:schemeClr>
        </a:gs>
        <a:gs pos="70000">
          <a:srgbClr val="C4D6EB"/>
        </a:gs>
        <a:gs pos="100000">
          <a:schemeClr val="bg1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
          </a:gs>
        </a:gsLst>
      </a:gradFill>
    </pic:spPr>
    …
  </pic:pic>
```

The above picture object is shown as an example of this double fill kind. *End example*]

Attributes	Description
dpi (DPI Setting) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the DPI (dots per inch) used to calculate the size of the blip. If not present or zero, the DPI in the blip is used. [*Note*: This attribute is primarily used to keep track of the picture quality within a document. There are different levels of quality needed for print than on-screen viewing and thus a need to track this information. *end note*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
rotWithShape (Rotate With Shape) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the fill should rotate with the shape. That is, when the shape that has been filled with a picture and the containing shape (say a rectangle) is transformed with a rotation then the fill is transformed with the same rotation. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BlipFillProperties) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.3.2.3 cNvCxnSpPr (Non-Visual Connection Shape Drawing Properties)

This element specifies the non-visual drawing properties for a connector shape. These non-visual properties are properties that the generating application would utilize when rendering the parent chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualConnectorProperties) is located in §A.4.1. *end note*]

21.3.2.4 cNvGraphicFramePr (Non-Visual Graphic Frame Drawing Properties)

This element specifies the non-visual drawing properties for a graphic frame. These non-visual properties are properties that the generating application would utilize when rendering the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGraphicFrameProperties) is located in §A.4.1. *end note*]

21.3.2.5 cNvGrpSpPr (Non-Visual Group Shape Drawing Properties)

This element specifies the non-visual drawing properties for a group shape. These non-visual properties are properties that the generating application would utilize when rendering the chart.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualGroupDrawingShapeProps) is located in §A.4.1. *end note*]

21.3.2.6 cNvPicPr (Non-Visual Picture Drawing Properties)

This element specifies the non-visual properties for the picture canvas. These properties are to be used by the generating application to determine how certain properties are to be changed for the picture object in question.

[*Example*: Consider the following DrawingML

```
<pic:pic>
  …
  <pic:nvPicPr>
    <pic:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
    <pic:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </pic:cNvPicPr>
    <pic:nvPr/>
  </pic:nvPicPr>
  …
</pic:pic>
```

End example]

Attributes	Description
preferRelativeResize (Relative Resize	Specifies if the user interface should show the resizing of the picture based on the picture's current size or its original size. If this attribute is set to true, then scaling is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Preferred) Namespace: http://purl.oclc.org/ooxml/drawingml/main	relative to the original picture size as opposed to the current picture size. [*Example*: Consider the case where a picture has been resized within a document and is now 50% of the originally inserted picture size. Now if the user chooses to make a later adjustment to the size of this picture within the generating application, then the value of this attribute should be checked. If this attribute is set to true then a value of 50% is shown. Similarly, if this attribute is set to false, then a value of 100% should be shown because the picture has not yet been resized from its current (smaller) size. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualPictureProperties) is located in §A.4.1. *end note*]

21.3.2.7 cNvPr (Non-Visual Drawing Properties)

This element specifies non-visual canvas properties. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following ChartDrawingML.

```
<cdr:pic>
  …
  <cdr:nvPicPr>
    <cdr:cNvPr id="4" name="Lilly_by_Lisher.jpg"/>
  </cdr:nvPicPr>
  …
</cdr:pic>
```

end example]

Attributes	Description
descr (Alternative Text for Object) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which do not display the current object. If this element is omitted, then no alternative text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… descr="A picture of a bowl of fruit">` The descr attribute contains alternative text which can be used in place of the actual

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
hidden (Hidden) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies whether this DrawingML object is displayed. When a DrawingML object is displayed within a document, that object can be hidden (i.e., present, but not visible). This attribute determines whether the object is rendered or made hidden. [*Note*: An application can have settings which allow this object to be viewed. *end note*] If this attribute is omitted, then the parent DrawingML object shall be displayed (i.e., not hidden). [*Example*: Consider an inline DrawingML object which must be hidden within the document's content. This setting would be specified as follows: `<… hidden="true" />` The hidden attribute has a value of `true`, which specifies that the DrawingML object is hidden and not displayed when the document is displayed. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
id (Unique Identifier) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a unique identifier for the current DrawingML object within the current document. This ID can be used to assist in uniquely identifying this object so that it can be referred to by other parts of the document. If multiple objects within the same document share the same id attribute value, then the document shall be considered non-conformant. [*Example*: Consider a DrawingML object defined as follows: `<… id="10" … >` The id attribute has a value of `10`, which is the unique identifier for this DrawingML object. *end example*] The possible values for this attribute are defined by the ST_DrawingElementId simple type (§20.1.10.21).
name (Name) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the name of the object. [*Note*: Typically, this is used to store the original file name of a picture object. *end note*] [*Example*: Consider a DrawingML object defined as follows: `< … name="foo.jpg" >`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The name attribute has a value of `foo.jpg`, which is the name of this DrawingML object. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
title (Title) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the title (caption) of the current DrawingML object. If this attribute is omitted, then no title text is present for the parent object. [*Example*: Consider a DrawingML object defined as follows: `<… title="Process Flow Diagram">` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingProps) is located in §A.4.1. *end note*]

21.3.2.8 cNvSpPr (Non-Visual Shape Drawing Properties)

This element specifies the non-visual drawing properties for a shape. These properties are to be used by the generating application to determine how the shape should be dealt with.

[*Example*: Consider the shape that has a shape lock applied to it.

```
<cdr:sp>
  <cdr:nvSpPr>
    <cdr:cNvPr id="2" name="Rectangle 1"/>
    <cdr:cNvSpPr>
      <a:spLocks noGrp="1"/>
    </cdr:cNvSpPr>
  </cdr:nvSpPr>
  …
</cdr:sp>
```

This shape lock is stored within the non-visual drawing properties for this shape. *end example*]

Attributes	Description
txBox (Text Box) Namespace:	Specifies that the corresponding shape is a text box and thus should be treated as such by the generating application. If this attribute is omitted then it is assumed that the corresponding shape is not specifically a text box.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
http://purl.oclc.org/ooxml/drawingml/main	[*Note*: Because a shape is not specified to be a text box does not mean that it cannot have text attached to it. A text box is merely a specialized shape with specific properties. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NonVisualDrawingShapeProps) is located in §A.4.1. *end note*]

21.3.2.9 cxnSp (Connection Shape)

This element specifies a connection shape that is used to connect two sp elements. Once a connection is specified using a cxnSp, it is left to the generating application to determine the exact path the connector takes. That is the connector routing algorithm is left up to the generating application as the desired path might be different depending on the specific needs of the application.

[*Example*:

end example]

[*Example*: Consider the following connector shape that connects two regular shapes.

```
<cdrcdr:grpSp>
  …
  <cdr:sp>
    <cdr:nvSpPr>
      <cdr:cNvPr id="1" name="Rectangle 1"/>
      <cdr:cNvSpPr/>
      <cdr:nvPr/>
    </cdr:nvSpPr>
    …
  </cdr:sp>
  <cdr:sp>
    <cdr:nvSpPr>
      <cdr:cNvPr id="2" name="Rectangle 2"/>
      <cdr:cNvSpPr/>
      <cdr:nvPr/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      </cdr:nvSpPr>
      …
    </cdr:sp>
    <cdr:cxnSp>
      <cdr:nvCxnSpPr>
        <cdr:cNvPr id="3" name="Elbow Connector 3"/>
        <cdr:cNvCxnSpPr>
          <a:stCxn id="1" idx="3"/>
          <a:endCxn id="2" idx="1"/>
        </cdr:cNvCxnSpPr>
        <cdr:nvPr/>
      </cdr:nvCxnSpPr>
      …
    </cdr:cxnSp>
  </cdr:grpSp>
```

end example]

Attributes	Description

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`..`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Connector) is located in §A.5.2. *end note*]

21.3.2.10 ext (Shape Extent)

This element describes the length and width properties for how far a drawing element should extend for.

Attributes	Description
cx (Extent Length) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the length of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `<… cx="1828800" cy="200000"/>` The cx attributes specifies that this object has a height of 1828800 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
cy (Extent Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the width of the extents rectangle in EMUs. This rectangle shall dictate the size of the object as displayed (the result of any scaling to the original object). [*Example*: Consider a DrawingML object specified as follows: `< … cx="1828800" cy="200000"/>` The cy attribute specifies that this object has a width of 200000 EMUs (English Metric Units). *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PositiveSize2D) is located in §A.4.1. *end note*]

21.3.2.11 from (Starting Anchor Point)

This element specifies the first anchor point for the drawing element. This is used to anchor the top and left sides of the shape within the chart. That is when the corresponding chart is adjusted, the shape is also adjusted.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following Chart Drawing content:

```
<cdr:relSizeAnchor>
  <cdr:from>
    <cdr:x>0.04583</cdr:x>
    <cdr:y>0.53125</cdr:y>
  </cdr:from>
  <cdr:to>
    <cdr:x>0.24583</cdr:x>
    <cdr:y>0.86458</cdr:y>
  </cdr:to>
  <cdr:sp macro="" textlink="">
  …
  </cdr:sp>
</cdr:relSizeAnchor>
```

The above example shows the first anchor point being specified via the from element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.2. *end note*]

21.3.2.12 graphicFrame (Graphic Frame)

This element specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.

Attributes	Description

fPublished (Publish To Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicFrame) is located in §A.5.2. *end note*]

21.3.2.13 grpSp (Group Shape)

This element specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would. The idea behind grouping elements however is that a single transform can apply to many shapes at the same time.

[*Example*: Consider the following group shape.

```
<cdr:grpSp>
  <cdr:nvGrpSpPr>
    <cdr:cNvPr id="10" name="Group 9"/>
    <cdr:cNvGrpSpPr/>
    <cdr:nvPr/>
  </cdr:nvGrpSpPr>
  <cdr:grpSpPr>
    <a:xfrm>
      <a:off x="838200" y="990600"/>
      <a:ext cx="2426208" cy="978408"/>
      <a:chOff x="838200" y="990600"/>
      <a:chExt cx="2426208" cy="978408"/>
   </a:xfrm>
  </cdr:grpSpPr>
  <cdr:sp>
  …
  </cdr:sp>
  <cdr:sp>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    …
    </cdr:sp>
    <cdr:sp>
    …
    </cdr:sp>
  </cdr:grpSp>
```

In the above example we see three shapes specified within a single group. These three shapes have their position and sizes specified just as they normally would within the shape tree. The generating application should apply the transformation after the bounding box for the group shape has been calculated. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShape) is located in §A.5.2. *end note*]

21.3.2.14 grpSpPr (Group Shape Properties)

This element specifies the properties that are to be common across all of the shapes within the corresponding group. If there are any conflicting properties within the group shape properties and the individual shape properties then the individual shape properties should take precedence.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the group shape should be rendered using only black and white coloring. That is the coloring information for the group shape should be converted to either black or white when rendering the corresponding shapes. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the group shapes themselves are stored with only black and white color information. This attribute instead sets the rendering mode that the shapes use when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeProperties) is located in §A.4.1. *end note*]

21.3.2.15 nvCxnSpPr (Connector Non Visual Properties)

This element specifies all non-visual properties for a connection shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a connection shape. This allows for additional information that does not affect the appearance of the connection shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ConnectorNonVisual) is located in §A.5.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.3.2.16 nvGraphicFramePr (Non-Visual Graphic Frame Properties)

This element specifies all non-visual properties for a graphic frame. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a graphic frame. This allows for additional information that does not affect the appearance of the graphic frame to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GraphicFrameNonVisual) is located in §A.5.2. *end note*]

21.3.2.17 nvGrpSpPr (Non-Visual Group Shape Properties)

This element specifies all non-visual properties for a group shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a group shape. This allows for additional information that does not affect the appearance of the group shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupShapeNonVisual) is located in §A.5.2. *end note*]

21.3.2.18 nvPicPr (Non-Visual Picture Properties)

This element specifies the non visual properties for a picture. This allows for additional information that does not affect the appearance of the picture to be stored.

[*Example*: Consider the following DrawingML.

```
<pic:pic>
  …
  <pic:nvPicPr>
    …
  </pic:nvPicPr>
  …
</pic:pic>
```

End example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PictureNonVisual) is located in §A.5.2. *end note*]

21.3.2.19 nvSpPr (Non-Visual Shape Properties)

This element specifies all non-visual properties for a shape. This element is a container for the non-visual identification properties, shape properties and application properties that are to be associated with a shape. This allows for additional information that does not affect the appearance of the shape to be stored.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeNonVisual) is located in §A.5.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.3.2.20 pic (Picture)

This element specifies the existence of a picture object within the document.

[*Example*: Consider the following ChartDrawingML that specifies the existence of a picture within a document. This picture can have non-visual properties, a picture fill as well as shape properties attached to it.

```
<cdr:pic>
  <cdr:nvPicPr>
    <cdr:cNvPr id="4" name="lake.JPG" descr="Picture of a Lake" />
    <cdr:cNvPicPr>
      <a:picLocks noChangeAspect="1"/>
    </cdr:cNvPicPr>
    <cdr:nvPr/>
  </cdr:nvPicPr>
  <cdr:blipFill>
  …
  </cdr:blipFill>
  <cdr:spPr>
  …
  </cdr:spPr>
</cdr:pic>
```

end example]

Attributes	Description

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	understood. [*Example*: `<… macro="DoWork()" >` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Picture) is located in §A.5.2. *end note*]

21.3.2.21 relSizeAnchor (Relative Anchor Shape Size)

This element specifies that the shape described here to reside within a chart should be sized based on relative anchor points. This is achieved via two elements. The from element specifies the top left corner of the shape bounding box in a RTL(right-to-left) implementation. The to element then specifies the bottom right corner of the shape bounding box in a RTL(right-to-left) implementation and thus the size of the shape.

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelSizeAnchor) is located in §A.5.2. *end note*]

21.3.2.22 sp (Shape)

This element specifies the existence of a single shape. A shape can either be a preset or a custom geometry, defined using the DrawingML framework. In addition to geometry, each shape can have both visual and non-visual properties attached. Text and corresponding styling information can also be attached to a shape. This shape is specified along with all other shapes within either the shape tree or group shape elements.

Attributes	Description

fLocksText (Lock Text)	Specifies whether to allow for the editing of text within this shape when the worksheet on which the shape resides has been protected as defined by SpreadsheetML. This allows for the specifying of locked or "protected" text on a per-shape basis within a spreadsheet document. If this attribute is not specified then a value of 0, or false is assumed.
	
	[*Example*: Consider the following shape that does not have locked text on.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fLocksText="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

fPublished (Publish to Server)	Specifies whether the shape shall be published with the worksheet when sent to the spreadsheet server. This is for use when interfacing with a document server.
	
	[*Example*: Consider the following shape that is not published with the worksheet when it is published back on the spreadsheet server.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp fPublished="0">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

macro (Reference to Custom Function)	This element specifies the custom function associated with the chart. [*Example*: A macro script, add-in function, and so on. *end example*] The format of this string shall be application-defined, and should be ignored if not understood. [*Example*: `<… macro="DoWork()" >` *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

textlink (Text Link)	Specifies whether the text contained within this shape is linked to a cell within the spreadsheet. That is the text within the shape has the value defined in the referenced spreadsheet cell.
	
	[*Example*: Consider the following shape with text linked to cell A1.
	`<cdr:relSizeAnchor>`
	`…`
	`<cdr:sp macro="" textlink="A1">`
	`…`
	`</cdr:sp>`
	`…`
	`</cdr:relSizeAnchor>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	end example] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.5.2. *end note*]

21.3.2.23 spPr (Shape Properties)

This element specifies the visual shape properties that can be applied to a special shape such as a connector shape or picture. These are the same properties that are allowed to describe the visual properties of a shape but are used here to describe additional object-specific properties within a document. This allows for these shapes to have both the properties of a shape as well as specific properties that are unique to only them.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.3.2.24 style (Shape Style)

The element specifies the style that is applied to a shape and the corresponding references for each of the style components such as lines and fills.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.3.2.25 to (Ending Anchor Point)

This element specifies the second anchor point for the drawing element. This is used to anchor the bottom and right sides of the shape within the spreadsheet. That is when the corresponding chart is adjusted, the shape is also adjusted.

[*Example*: Consider the following ChartDrawingML

```
<cdr:relSizeAnchor>
  <cdr:from>
    <cdr:x>0.04583</cdr:x>
    <cdr:y>0.53125</cdr:y>
  </cdr:from>
  <cdr:to>
    <cdr:x>0.24583</cdr:x>
    <cdr:y>0.86458</cdr:y>
  </cdr:to>
  <cdr:sp macro="" textlink="">
  …
  </cdr:sp>
</cdr:relSizeAnchor>
```

The above example shows the second anchor point being specified via the to element. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Marker) is located in §A.5.2. *end note*]

21.3.2.26 txBody (Shape Text Body)

This element specifies the existence of text to be contained within the corresponding shape. All visible text and visible text related properties are contained within this element. There can be multiple paragraphs and within paragraphs multiple runs of text.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.3.2.27 x (Relative X Coordinate)

This element specifies the relative x coordinate that is used to define the percentage-based horizontal position for a shape within a chart drawing object. The coordinate boundaries are specified within the corresponding simple type listed below.

The possible values for this element are defined by the ST_MarkerCoordinate simple type (§21.3.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.3.2.28 xfrm (Graphic Frame Transform)

This element specifies a 2-D transform to be applied to a Graphic Frame.

Attributes	Description
flipH (Horizontal Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a horizontal flip. When true, this attribute defines that the shape is flipped horizontally about the center of its bounding box. [*Example*: The following illustrates the effect of a horizontal flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
flipV (Vertical Flip) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies a vertical flip. When true, this attribute defines that the group is flipped vertically about the center of its bounding box. [*Example*: The following illustrates the effect of a vertical flip. *end example*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies the rotation of the Graphic Frame. The units for which this attribute is specified in reside within the simple type definition referenced below. The possible values for this attribute are defined by the ST_Angle simple type (§20.1.10.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Transform2D) is located in §A.4.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.3.2.29 y (Relative Y Coordinate)

This element specifies the relative y coordinate that is used to define the percentage-based vertical position for a shape within a chart drawing object. The coordinate boundaries are specified within the corresponding simple type listed below.

The possible values for this element are defined by the ST_MarkerCoordinate simple type (§21.3.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

21.3.3 Simple Types

This is the complete list of simple types dedicated to DrawingML – Chart Drawings.

21.3.3.1 ST_MarkerCoordinate (Chart Marker Coordinate Value)

This simple type specifies the chart marker coordinate value. It is to be represented as a fractional position between 0.0 and 1.0 of the chart width or height with 0.0 being the left or top edge.

This simple type's contents are a restriction of the W3C XML Schema double datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.0.
- This simple type has a maximum value of less than or equal to 1.0.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_MarkerCoordinate) is located in §A.5.2. *end note*]

21.4 DrawingML - Diagrams

A DrawingML diagram allows the definition of diagrams using DrawingML objects and constructs. This namespace defines the contents of a DrawingML diagram.

21.4.1 Table of Contents

This subclause is informative.

21.4.2 Diagram Definition 3494
21.4.2.1 adj (Shape Adjust) 3494
21.4.2.2 adjLst (Shape Adjust List) 3494
21.4.2.3 alg (Algorithm) 3495
21.4.2.4 cat (Category) 3495
21.4.2.5 catLst (Category List) 3496
21.4.2.6 choose (Choose Element) 3496
21.4.2.7 clrData (Color Transform Sample Data) 3497
21.4.2.8 constr (Constraint) 3498
21.4.2.9 constrLst (Constraint List) 3501
21.4.2.10 dataModel (Data Model) 3502

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.4.2.11 desc (Description) 3502
21.4.2.12 else (Else) 3503
21.4.2.13 extLst (Extension List) 3504
21.4.2.14 forEach (For Each) 3504
21.4.2.15 if (If) 3506
21.4.2.16 layoutDef (Layout Definition) 3509
21.4.2.17 layoutDefHdr (Layout Definition Header) 3510
21.4.2.18 layoutDefHdrLst (Diagram Layout Header List) 3510
21.4.2.19 layoutNode (Layout Node) 3511
21.4.2.20 param (Parameter) 3512
21.4.2.21 presOf (Presentation Of) 3512
21.4.2.22 relIds (Explicit Relationships to Diagram Parts) 3514
21.4.2.23 resizeHandles (Shape Resize Style) 3515
21.4.2.24 rule (Rule) 3515
21.4.2.25 ruleLst (Rule List) 3517
21.4.2.26 sampData (Sample Data) 3518
21.4.2.27 shape (Shape) 3519
21.4.2.28 style (Shape Style) 3520
21.4.2.29 styleData (Style Data) 3520
21.4.2.30 title (Title) 3521
21.4.2.31 varLst (Variable List) 3521

21.4.3 Data 3522

21.4.3.1 bg (Background Formatting) 3522
21.4.3.2 cxn (Connection) 3522
21.4.3.3 cxnLst (Connection List) 3525
21.4.3.4 prSet (Property Set) 3526
21.4.3.5 pt (Point) 3529
21.4.3.6 ptLst (Point List) 3531
21.4.3.7 spPr (Shape Properties) 3531
21.4.3.8 t (Text Body) 3531
21.4.3.9 whole (Whole E2O Formatting) 3532

21.4.4 Color Information 3532

21.4.4.1 cat (Color Transform Category) 3532
21.4.4.2 catLst (Color Transform Category List) 3532
21.4.4.3 colorsDef (Color Transform Definitions) 3533
21.4.4.4 colorsDefHdr (Color Transform Definition Header) 3534
21.4.4.5 colorsDefHdrLst (Color Transform Header List) 3535
21.4.4.6 desc (Description) 3535
21.4.4.7 effectClrLst (Effect Color List) 3535
21.4.4.8 fillClrLst (Fill Color List) 3537
21.4.4.9 linClrLst (Line Color List) 3538
21.4.4.10 styleLbl (Style Label) 3539
21.4.4.11 title (Title) 3540
21.4.4.12 txEffectClrLst (Text Effect Color List) 3540
21.4.4.13 txFillClrLst (Text Fill Color List) 3541
21.4.4.14 txLinClrLst (Text Line Color List) 3542

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.5 Style Definitions 3543
21.4.5.1 cat (Category) 3543
21.4.5.2 catLst (Category List) 3544
21.4.5.3 desc (Style Label Description) 3544
21.4.5.4 presLayoutVars (Presentation Layout Variables) 3545
21.4.5.5 scene3d (3-D Scene) 3545
21.4.5.6 sp3d (3-D Shape Properties) 3545
21.4.5.7 styleDef (Style Definition) 3547
21.4.5.8 styleDefHdr (Style Definition Header) 3549
21.4.5.9 styleDefHdrLst (List of Style Definition Headers) 3550
21.4.5.10 styleLbl (Style Label) 3550
21.4.5.11 title (Title) 3551
21.4.5.12 txPr (Text Properties) 3551

21.4.6 Layout Definition 3552
21.4.6.1 animLvl (Level Animation) 3552
21.4.6.2 animOne (One by One Animation String) 3552
21.4.6.3 bulletEnabled (Show Insert Node) 3553
21.4.6.4 chMax (Maximum Children) 3553
21.4.6.5 chPref (Preferred Number of Children) 3554
21.4.6.6 dir (Diagram Direction) 3555
21.4.6.7 hierBranch (Organization Chart Branch Style) 3555
21.4.6.8 orgChart (Show Organization Chart User Interface) 3555

21.4.7 Simple Types 3556
21.4.7.1 ST_AlgorithmType (Algorithm Types) 3556
21.4.7.2 ST_AnimLvlStr (Animation Level String Definition) 3557
21.4.7.3 ST_AnimOneStr (One by One Animation Value Definition) 3558
21.4.7.4 ST_ArrowheadStyle (Arrowhead Styles) 3558
21.4.7.5 ST_AutoTextRotation (Auto Text Rotation) 3558
21.4.7.6 ST_AxisType (Axis Type) 3559
21.4.7.7 ST_AxisTypes (Axis Type List) 3565
21.4.7.8 ST_BendPoint (Bend Point) 3565
21.4.7.9 ST_Booleans (Boolean List.) 3565
21.4.7.10 ST_BoolOperator (Boolean Constraint) 3565
21.4.7.11 ST_Breakpoint (Breakpoint) 3566
21.4.7.12 ST_CenterShapeMapping (Center Shape Mapping) 3566
21.4.7.13 ST_ChildAlignment (Child Alignment) 3567
21.4.7.14 ST_ChildDirection (Child Direction) 3567
21.4.7.15 ST_ChildOrderType (Child Order) 3567
21.4.7.16 ST_ClrAppMethod (Color Application Method Type) 3568
21.4.7.17 ST_ConnectorDimension (Connector Dimension) 3570
21.4.7.18 ST_ConnectorPoint (Connector Point) 3570
21.4.7.19 ST_ConnectorRouting (Connector Routing) 3571
21.4.7.20 ST_ConstraintRelationship (Constraint Relationship) 3571
21.4.7.21 ST_ConstraintType (Constraint Type) 3572
21.4.7.22 ST_ContinueDirection (Continue Direction) 3574
21.4.7.23 ST_CxnType (Connection Type) 3575

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.4.7.24 ST_DiagramHorizontalAlignment (Horizontal Alignment) 3575
21.4.7.25 ST_DiagramTextAlignment (Text Alignment) 3576
21.4.7.26 ST_Direction (Diagram Direction Definition) 3576
21.4.7.27 ST_ElementType (Data Point Type) 3577
21.4.7.28 ST_ElementTypes (Diagram Layout Node Type List) 3577
21.4.7.29 ST_FallbackDimension (Fallback Dimension) 3577
21.4.7.30 ST_FlowDirection (Flow Direction) 3578
21.4.7.31 ST_FunctionArgument (Function Argument) 3578
21.4.7.32 ST_FunctionOperator (Function Operator) 3578
21.4.7.33 ST_FunctionType (Function Type) 3579
21.4.7.34 ST_FunctionValue (Function Value) 3579
21.4.7.35 ST_GrowDirection (Grow Direction) 3580
21.4.7.36 ST_HierarchyAlignment (Hierarchy Alignment) 3580
21.4.7.37 ST_HierBranchStyle (Hierarchy Branch Style Definition) 3582
21.4.7.38 ST_HueDir (Hue Direction) 3583
21.4.7.39 ST_Index1 (1-Based Index) 3584
21.4.7.40 ST_Ints (Integer List) 3584
21.4.7.41 ST_LayoutShapeType (Layout Shape Type) 3584
21.4.7.42 ST_LinearDirection (Linear Direction) 3584
21.4.7.43 ST_ModelId (Model Identifier) 3585
21.4.7.44 ST_NodeCount (Number of Nodes Definition) 3585
21.4.7.45 ST_NodeHorizontalAlignment (Node Horizontal Alignment) 3585
21.4.7.46 ST_NodeVerticalAlignment (Node Vertical Alignment) 3586
21.4.7.47 ST_Offset (Offset) 3586
21.4.7.48 ST_OutputShapeType (Output Shape Type) 3586
21.4.7.49 ST_ParameterId (Parameter Identifier) 3587
21.4.7.50 ST_ParameterVal (Parameter Values) 3590
21.4.7.51 ST_PtType (Point Type) 3591
21.4.7.52 ST_PyramidAccentPosition (Pyramid Accent Position) 3592
21.4.7.53 ST_PyramidAccentTextMargin (Pyramid Accent Text Margin) 3592
21.4.7.54 ST_ResizeHandlesStr (Resize Handle) 3593
21.4.7.55 ST_RotationPath (Rotation Path) 3594
21.4.7.56 ST_SecondaryChildAlignment (Secondary Child Alignment) 3594
21.4.7.57 ST_SecondaryLinearDirection (Secondary Linear Direction) 3595
21.4.7.58 ST_StartingElement (Starting Element) 3595
21.4.7.59 ST_TextAnchorHorizontal (Text Anchor Horizontal) 3596
21.4.7.60 ST_TextAnchorVertical (Text Anchor Vertical) 3596
21.4.7.61 ST_TextBlockDirection (Text Block Direction) 3596
21.4.7.62 ST_TextDirection (Text Direction) 3597
21.4.7.63 ST_UnsignedInts (Unsigned Integer List) 3597
21.4.7.64 ST_VariableType (Variable Type) 3597
21.4.7.65 ST_VerticalAlignment (Vertical Alignment) 3598
21.4.7.66 ST_PrSetCustVal (Property Set Customized Value) 3598

End of informative text.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.2 Diagram Definition

This section specifies the elements which define the layout and hierarchy of a diagram based on its constituent nodes and connections.

21.4.2.1 adj (Shape Adjust)

Shape adjust value. These can be used to modify the adjust handles supported on various auto shapes. It is only possible to set the initial value, not to modify it using constraints and rules.

[*Example*: Consider the following example of the adj element in a DrawingML diagram:

```
<adjLst>
    <adj idx="2" val=".35" />
</adjLst>
```

In this example we have a single adjust handle being modified by setting its value to `0.35`. *end example*]

Attributes	Description
idx (Adjust Handle Index)	Adjust value index. Different shapes support different adjust handles. The possible values for this attribute are defined by the ST_Index1 simple type (§21.4.7.39).
val (Value)	An absolute value. The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Adj) is located in §A.5.3. *end note*]

21.4.2.2 adjLst (Shape Adjust List)

This element is simply a list of shape adjusts.

[*Example*: Consider the following example of the adjLst element in a DrawingML diagram:

```
<adjLst>
   <adj idx="1" val="1.35" />
    <adj idx="2" val=".35" />
</adjLst>
```

In this example we have a two adjust handle being modified for the containing shape. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdjLst) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.4.2.3 alg (Algorithm)

The algorithm used by the containing layout node. The algorithm defines the behavior of the layout node along with the behavior and layout of the nested layout nodes.

[*Example*: Consider the following example of alg being used in a DrawingML diagram:

```
<layoutNode name="arrow">
    <varLst/>
    <alg type="tx" />
    <shape type="upArrow">
          <adjLst>
                <adj idx="2" val=".35" />
          </adjLst>
    </shape>
    <presOf axis="desOrSelf" ptType="node" />
    <ruleLst/>
</layoutNode>
```

In this example, the `tx` algorithm is being used to layout text within the containing layout node. *end example*]

Attributes	Description
rev (Revision Number)	The revision number of an algorithm. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Algorithm Type)	Specifies the algorithm type. The possible values for this attribute are defined by the ST_AlgorithmType simple type (§21.4.7.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Algorithm) is located in §A.5.3. *end note*]

21.4.2.4 cat (Category)

This element specifies a category in the user interface where this layout definition displays to the user.

[*Example*: Consider the following example of a cat in a DrawingML diagram:

```
<catLst>
    <cat type="relationship" pri="19000" />
</catLst>
```

In this example we define a single category called `relationship` which has a priority of `19000`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
pri (Priority)	The priority within the category for this diagram determines the order in which it displays in the user interface. Lower numbers are displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	Specifies the category type associated with the element. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Category) is located in §A.5.3. *end note*]

21.4.2.5 catLst (Category List)

This element is simply a list of cat elements.

[*Example*: Consider the following example of a catLst element in a DrawingML diagram:

```
<catLst>
    <cat type="list" pri="18000" />
    <cat type="relationship" pri="19000" />
</catLst>
```

In this example we define two different categories which are to be displayed in the user interface. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Categories) is located in §A.5.3. *end note*]

21.4.2.6 choose (Choose Element)

The choose element wraps if/else blocks into a choose block.

[*Example*: Consider the following example of a choose element in a DrawingML diagram:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
        <alg type="snake">
            <param type="grDir" val="tL"/>
            <param type="flowDir" val="row"/>
            <param type="contDir" val="sameDir"/>
            <param type="off" val="ctr"/>
        </alg>
    </if>
    <else name="Name3">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <alg type="snake">
                  <param type="grDir" val="tR"/>
                  <param type="flowDir" val="row"/>
                  <param type="contDir" val="sameDir"/>
                  <param type="off" val="ctr"/>
            </alg>
      </else>
</choose>
```

In this example, a choose element is used to define two different sets of parameters associated with a snake algorithm depending upon the direction in which the user wants the algorithm to flow (RTL or LTR). *end example*]

Attributes	Description

name (Name)	A unique name associated with the choose statement.
	
	[*Example*: Consider the following example of a choose element in a DrawingML diagram:
	
	`<choose name="Name1">`
	
	…
	
	`</choose>`
	
	In this example, the choose element is named `Name1`. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Choose) is located in §A.5.3. *end note*]

21.4.2.7 clrData (Color Transform Sample Data)

This element defines the sample data that is to be used in the user interface controls regarding displaying color transforms for a given diagram. This sample data predefines a data model to be combined with a layout definition in order to create a diagram which a color transform can be applied and displayed to the user as an example of the color transform.

[*Example*: Consider the following example of a clrData element in a DrawingML diagram:

```
<clrData>
    <dataModel>
          <ptLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
                <pt modelId="0" type="doc"/>
                <pt modelId="1"/>
                <pt modelId="2"/>
                <pt modelId="3"/>
                <pt modelId="4"/>
                <pt modelId="5"/>
                <pt modelId="6"/>
            </ptLst>
            <cxnLst>
                <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
                <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
                <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
                <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
                <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
            </cxnLst>
            <bg/>
            <whole/>
        </dataModel>
    </clrData>
```

In this example we define 6 points which all connect back to a seventh document type point. *end example*]

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SampleData) is located in §A.5.3. *end note*]

21.4.2.8 constr (Constraint)

This element is used to specify size, position of nodes, text values, and layout dependencies between nodes in a layout definition.

[*Example*: Consider the following example of a constraint list which contains some example constraints being defined and applied to layout nodes in the layout definition:

```
    <constrLst>
        <constr type="w" for="ch" forName="node1" refType="w" refForName=""/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <constr type="h" for="ch" forName="node1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.6"/>
    <constr type="w" for="ch" forName="transition1" refType="w" refFor="ch"
refForName="node1" op="equu" fact="0.1"/>
    <constr type="primFontSz" for="ch" forName="node1" refForName="" op="equ"
val="100"/>
</constrLst>
```

In this example we can see constraints being defined for the width and height along with the primary font size for a layout node referenced by `node1`. The width for a transition is also specified. *end example*]

Attributes	Description
fact (Factor)	Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. [*Example*: Consider the following example of fact in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the width for `transition1` is being defined as one-tenth the width of `node1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.
for (For)	Specifies the axis of layout nodes to apply a constraint or rule to. [*Example*: Consider the following example of for in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the for attribute is specifying that `node1` is a child node to the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
forName (For Name)	Specifies the name of the layout node to apply a constraint or rule to. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, forName is specifying the layout node named `transition1` for its reference. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
op (Operator)	The operator constraint used to evaluate the condition. [*Example*: Consider the following example of op in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, op is specifying an equality defined between the two referencing values. *end example*] The possible values for this attribute are defined by the ST_BoolOperator simple type (§21.4.7.10).
ptType (Data Point Type)	Specifies the type of data point to select. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
refFor (Reference For)	The for value of the referenced constraint. [*Example*: Consider the following example of refFor in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refFor is specifying the reference constraint is a child of the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
refForName (Reference For Name)	The name of the layout node referenced by a reference constraint. [*Example*: Consider the following example of refForName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refForName is specifying the layout node named node1 for its reference. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
refPtType (Reference Point Type)	The point type used int he referenced constraint. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
refType (Reference Type)	Specifies the type of a reference constraint. [*Example*: Consider the following example of refType in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, refType is specifying referencing the width attribute of `node1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
type (Constraint Type)	Specifies the constraint to apply to this layout node. [*Example*: Consider the following example of type in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, type is specifying the width attribute of `transition1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
val (Value)	Specifies an absolute value instead of reference another constraint. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" val="10"/>` In this example, val is specifying the absolute value of the width of `transition1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Constraint) is located in §A.5.3. *end note*]

21.4.2.9 constrLst (Constraint List)

This element is simply a list of constraints.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following example of a constraint list which contains some example constraints which are being defined and applied to layout nodes in the layout definition:

```
<constrLst>
    <constr type="w" for="ch" forName="node1" refType="w" refForName=""/>
    <constr type="h" for="ch" forName="node1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.6"/>
    <constr type="w" for="ch" forName="transition1" refType="w" refFor="ch"
refForName="node1" op="equ" fact="0.1"/>
    <constr type="primFontSz" for="ch" forName="node1" refForName="" op="equ"
val="100"/>
</constrLst>
```

In this example we can see constraints being defined for the width and height along with the primary font size for a layout node referenced by `node1`. The width for a transition is also specified. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Constraints) is located in §A.5.3. *end note*]

21.4.2.10 dataModel (Data Model)

The data for this instance of the diagram. Either a sample data model, or the data the user has entered.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DataModel) is located in §A.5.3. *end note*]

21.4.2.11 desc (Description)

This element holds a description for a layout definition. The description can be used to describe the qualities associated with a particular layout definition.

Attributes	Description
lang (Language)	The natural language of the title or description of this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Value)	The string which is used as the description of the layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Description) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.4.2.12 else (Else)

This element is similar to an else statement in a programming language in that it wraps elements which are to be used when the if conditionals are not true.

[*Example*: Consider the following example of an else element in a DrawingML diagram within the context of a choose statement:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
          <alg type="snake">
                <param type="grDir" val="tL"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </if>
    <else name="Name3">
          <alg type="snake">
                <param type="grDir" val="tR"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </else>
</choose>
```

In this example, a else element is used to define a set of parameters associated with the snake algorithm when the diagram is reversed. *end example*]

Attributes	Description

name (Name)	A unique name associated with the choose statement.
	
	[*Example*: Consider the following example of a else element in a DrawingML diagram:
	
	`<else name="Name1">`
	
	…
	
	`</else>`
	
	In this example, the else element is named `Name1`. *end example*].
	
	The possible values for this attribute are defined by the W3C XML Schema string

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Otherwise) is located in §A.5.3. *end note*]

21.4.2.13 extLst (Extension List)

This element specifies an extension list, within which all future extensions are defined within ext elements.

The extension list along with corresponding future extensions is used to extend the storage capabilities of the DrawingML framework. This allows for various new types of data to be stored natively within the existing diagram syntax.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OfficeArtExtensionList) is located in §A.4.1. *end note*]

21.4.2.14 forEach (For Each)

A looping structure, similar to a `for` loop in a programming language, which defines what data model points use this layout node.

[*Example*: Consider the following example of a forEach being used within a DrawingML diagram:

```
<forEach name="Name5" ref="" axis="ch" ptType="node">
    <layoutNode name="node1" styleLbl="" moveWith="">
          <alg type="sp"/>
          <shape
xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships" r:blip="">
                <adjLst/>
          </shape>
          <constrLst/>
    </layoutNode>
</forEach>
```

In this example, the forEach element creates a layout node, referenced by the name `node1`, for every associated data model point in the diagram. In this particular instance the forEach creates the layout node for every child of the current point node. *end example*]

Attributes	Description
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type (§21.4.7.9).
name (Name)	A unique identifier for the layout node. The possible values for this attribute are defined by the W3C XML Schema string datatype.

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

ref (Reference)	When used on a for-each element, causes the specified for-each element to be used instead. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram: `<presOf axis="desOrSelf" ptType="node" st="2"/>` In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ForEach) is located in §A.5.3. *end note*]

21.4.2.15 if (If)

Like an if statement in a programming language, wraps elements which are to be used under the conditions defined by its attributes.

[*Example*: Consider the following example of an if element in a DrawingML diagram within the context of a choose statement:

```
<choose name="Name1">
    <if name="Name2" func="var" arg="dir" op="equ" val="norm">
          <alg type="snake">
                <param type="grDir" val="tL"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </if>
    <else name="Name3">
          <alg type="snake">
                <param type="grDir" val="tR"/>
                <param type="flowDir" val="row"/>
                <param type="contDir" val="sameDir"/>
                <param type="off" val="ctr"/>
          </alg>
    </else>
</choose>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example, a if element is used to define a set of parameters associated with the snake algorithm when the diagram is in the normal direction. *end example*]

Attributes	Description
arg (Argument)	Specifies the variable to use as part of the test in an if element. Ignored unless the function attribute is set to "var". The possible values for this attribute are defined by the ST_FunctionArgument simple type (§21.4.7.31).
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*] The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

func (Function)	The function used to evaluate the if condition.
	
	[*Example*: Consider the following example of `func` being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	…
	`</if>`
	
	In this example, `func` is set to `var`. *end example*]
	
	The possible values for this attribute are defined by the ST_FunctionType simple type (§21.4.7.33).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§21.4.7.9).

name (Name)	A unique identifier for the layout node.
	
	The function used to evaluate the if condition.
	
	[*Example*: Consider the following example of name being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	`…`
	`</if>`
	
	In this example, the name attribute is set to Name2. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

op (Operator)	The operator used to evaluate the condition.
	
	[*Example*: Consider the following example of op being used in DrawingML:
	
	`<if name="Name2" func="var" arg="dir" op="equ" val="norm">`
	`…`
	`</if>`
	
	In this example, op is being used to test the equality of the argument and value. *end example*]
	
	The possible values for this attribute are defined by the ST_FunctionOperator simple type (§21.4.7.32).

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	`…`
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<presOf axis="desOrSelf" ptType="node" st="2"/> In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
val (Value)	An absolute value. The possible values for this attribute are defined by the ST_FunctionValue simple type (§21.4.7.34).

[*Note*: The W3C XML Schema definition of this element's content model (CT_When) is located in §A.5.3. *end note*]

21.4.2.16 layoutDef (Layout Definition)

This element is the root element for defining a layout definition. The layout definition is defined through a set of nested layout nodes. The layout definition is responsible for defining the look of a diagram.

Attributes	Description
defStyle (Default Style)	This attribute defines a reference to a default style which is to be applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
minVer (Minimum Version)	Minimum product version that can support this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
uniqueId (Unique Identifier)	The unique identifier for this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DiagramDefinition) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.2.17 layoutDefHdr (Layout Definition Header)

This element is the header information representing the minimum knowledge needed by an application to preload information about a layout definition. This preloading allows for the actual load of the layout definition to occur at a later time which helps with any performance concerns an application might have.

[*Example*: Consider the following example of a layoutDefHdr within a DrawingML diagram:

```
<layoutDefHdr uniqueId="urn:layout/default">
    <title val="Basic Block List" />
    <desc val="" />
    <catLst>
        <cat type="list" pri="1000" />
    </catLst>
</layoutDefHdr>
```

In this example we define a title along with a category and prioritization for the diagram referenced by the `uniqueId` of `urn:layout:default`. *end example*]

Attributes	Description
defStyle (Default Style)	This attribute defines a reference to a default style which is to be applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
minVer (Minimum Version)	Minimum product version that can support this Diagram Layout. The possible values for this attribute are defined by the W3C XML Schema string datatype.
resId (Resource Identifier)	Resource ID used internally. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique Identifier)	The unique identifier for this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DiagramDefinitionHeader) is located in §A.5.3. *end note*]

21.4.2.18 layoutDefHdrLst (Diagram Layout Header List)

This element is simply a list of layout definition headers. This list of headers is used internally as a way to group all of the layout definition headers together into a single structure.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_DiagramDefinitionHeaderLst) is located in §A.5.3. *end note*]

21.4.2.19 layoutNode (Layout Node)

The layout node is the basic building block of diagrams. The layout node is responsible for defining how shapes are arranged in a diagram and how the data maps to a particular shape in a diagram.

[*Example*: Consider the following example of a basic layout node defined in a DrawingML diagram:

```
<layoutNode name="node">
    <varLst>
          <bulletEnabled val="1"/>
    </varLst>
    <presOf axis="desOrSelf" ptType="node"/>
    <alg type="tx"/>
    <shape type="rect"
xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships" r:blip="">
          <adjLst/>
    </shape>
    <constrLst/>
    <ruleLst>
          <rule type="primFontSz" forName="" val="2" fact="NaN" max="NaN"/>
    </ruleLst>
</layoutNode>
```

In this example we define a layout node which holds text and is a rectangle. *end example*]

Attributes	Description
chOrder (Child Order)	Specifes the ordering of the child layout nodes for a given layout node. The possible values for this attribute are defined by the ST_ChildOrderType simple type (§21.4.7.15).
moveWith (Move With)	Reference to another layout node that this layout node moves with. The possible values for this attribute are defined by the W3C XML Schema string datatype.
name (Name)	A unique identifier for the layout node. The possible values for this attribute are defined by the W3C XML Schema string datatype.
styleLbl (Style Label)	Specify which formatting option from a style or color variation should be applied to this layout node. The possible values for this attribute are defined by the W3C XML Schema string

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutNode) is located in §A.5.3. *end note*]

21.4.2.20 param (Parameter)

The parameter element modifies the default behavior of an algorithm.

[*Example*: Consider the following example of a param being used in a DrawingML diagram:

```
<alg type="snake">
    <param type="grDir" val="tL"/>
    <param type="flowDir" val="row"/>
    <param type="contDir" val="sameDir"/>
    <param type="off" val="ctr"/>
</alg>
```

In this example we see the snake algorithm being utilized and four parameters being set which are associated with the snake algorithm and modify its behavior. *end example*]

Attributes	Description
type (Parameter Type)	Specifies the parameter which is being modified. The possible values for this attribute are defined by the ST_ParameterId simple type (§21.4.7.49).
val (Value)	Specifies the actual value to be given to the parameter type defined by the type attribute. The possible values for this attribute are defined by the ST_ParameterVal simple type (§21.4.7.50).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Parameter) is located in §A.5.3. *end note*]

21.4.2.21 presOf (Presentation Of)

This element specifies a particular data model point which is to be mapped to the containing layout node. This attribute is responsible for defining the mapping of data to the layout nodes in a diagram.

[*Example*: Consider the following example of presOf in use within a DrawingML diagram:

```
<presOf axis="desOrSelf" ptType="node"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example the presOf element is mapping to a particular data model point. *end example*]

Attributes	Description
axis (Axis)	Specifies the axis on which to select data from the data model. [*Example*: `axis="ch"` selects children of the current point node and `axis="des"` selects all descendants. *end example*] The possible values for this attribute are defined by the ST_AxisTypes simple type (§21.4.7.7).

cnt (Count)	Specifies the count of items to use in a data set.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, up to two children are obtained through this forEach. *end example*]
	
	The possible values for this attribute are defined by the ST_UnsignedInts simple type (§21.4.7.63).

hideLastTrans (Hide Last Transition)	In algorithms that support transitions, this attribute specifies that the last transition is not rendered. This allows for diagrams that start and end with a node. The possible values for this attribute are defined by the ST_Booleans simple type (§21.4.7.9).

ptType (Data Point Type)	Specifies the type of data point to select.
	
	[*Example*: Consider the following example of a forEach in a DrawingML diagram:
	
	`<forEach name="Name5" ref="" axis="ch" ptType="node" cnt="2">`
	
	…
	
	`</forEach>`
	
	In this example, the forEach selects all node type points in the set. *end example*]
	
	The possible values for this attribute are defined by the ST_ElementTypes simple type (§21.4.7.28).

st (Start)	Specifies where to start in a data set. [*Example*: Consider the following example of a forEach in a DrawingML diagram:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	<presOf axis="desOrSelf" ptType="node" st="2"/> In this example, the second element in the set is the first point returned. *end example*] The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).
step (Step)	Specifies the step to use in a data set. A step with a value of 2 returns every other item in the set. The possible values for this attribute are defined by the ST_Ints simple type (§21.4.7.40).

[*Note*: The W3C XML Schema definition of this element's content model (CT_PresentationOf) is located in §A.5.3. *end note*]

21.4.2.22 relIds (Explicit Relationships to Diagram Parts)

This element specifies the relationship IDs used to explicitly reference each of the four constituent parts of a DrawingML diagram:

- Diagram Colors (cs attribute)
- Diagram Data (dm attribute)
- Diagram Layout Definition (lo attribute)
- Diagram Style (qs attribute)

Attributes	Description
cs (Explicit Relationship to Diagram Colors Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Colors part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramColors or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
dm (Explicit Relationship to Diagram Data Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Data part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramData or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
lo (Explicit Relationship to Diagram Layout Definition Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Layout Definition part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramLayout or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
qs (Explicit Relationship to Style Definition Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID for the explicit relationship to the Diagram Style part used by this diagram. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramQuickStyle or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).

[*Note*: The W3C XML Schema definition of this element's content model (CT_RelIds) is located in §A.5.3. *end note*]

21.4.2.23 resizeHandles (Shape Resize Style)

This element defines the behavior when resizing shapes within a diagram. Because the size of the shape plays a large role in the overall layout of other nodes within the diagram, there are two ways resize can occur on a node.

Attributes	Description
val (Shape Resize Style Type)	Specifies the behavior for a shape when resizing shapes within a diagram. If this attribute is not specified, the default value shall be `rel`. The possible values for this attribute are defined by the ST_ResizeHandlesStr simple type (§21.4.7.54).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ResizeHandles) is located in §A.5.3. *end note*]

21.4.2.24 rule (Rule)

This element allows for a rule to be specified which changes the value of an existing constraint.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: Consider the following example of a rule in a DrawingML diagram:

```
<ruleLst>
    <rule type="primFontSz" val="2" />
</ruleLst>
```

In this example a rule is being defined that shrinks the primary font size down to a lower limit of 2pt font when the text no longer fits correctly in the layout node. *end example*]

Attributes	Description
fact (Factor)	Factor used in a reference constraint or a rule in order to modify a referenced value by the factor defined. [*Example*: Consider the following example of fact in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the width for `transition1` is being defined as one-tenth the width of `node1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.
for (For)	Specifies the axis of layout nodes to apply a constraint or rule to. [*Example*: Consider the following example of for in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, the for attribute is specifying that `node1` is a child node to the current layout node. *end example*] The possible values for this attribute are defined by the ST_ConstraintRelationship simple type (§21.4.7.20).
forName (For Name)	Specifies the name of the layout node to apply a constraint or rule to. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, forName is specifying the layout node named `transition1` for its reference. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
max (Max Value)	Sets the maximum value for a constraint so rules can no longer increase the constraint beyond that value. The possible values for this attribute are defined by the W3C XML Schema double datatype.
ptType (Data Point Type)	Specifies the type of data point to select. The possible values for this attribute are defined by the ST_ElementType simple type (§21.4.7.27).
type (Constraint Type)	Specifies the constraint to apply to this layout node. [*Example*: Consider the following example of type in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" refType="w" refFor="ch" refForName="node1" op="equ" fact="0.1"/>` In this example, type is specifying the width attribute of `transition1`. *end example*] The possible values for this attribute are defined by the ST_ConstraintType simple type (§21.4.7.21).
val (Value)	Specifies an absolute value instead of reference another constraint. [*Example*: Consider the following example of forName in use in a DrawingML diagram: `<constr type="w" for="ch" forName="transition1" val="10"/>` In this example, val is specifying the absolute value of the width of `transition1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema double datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NumericRule) is located in §A.5.3. *end note*]

21.4.2.25 ruleLst (Rule List)

This element is simply a list of rules.

This element allows for a rule to be specified which changes the value of an existing constraint.

[*Example*: Consider the following example of a ruleLst in a DrawingML diagram:

```
<ruleLst>
    <rule type="primFontSz" val="2" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
</ruleLst>
```

In this example a single rule is being defined in the ruleLst that shrinks the primary font size down to a lower limit of 2pt font when the text no longer fits correctly in the layout node. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rules) is located in §A.5.3. *end note*]

21.4.2.26 sampData (Sample Data)

This element defines the sample data model which is used to pre-populate a diagram with placeholder data in order for the diagram to display itself in the user interface which shows all of the available diagrams to a user.

[*Example*: Consider the following example of a sampData element within a DrawingML diagram:

```
<sampData>
    <dataModel>
          <ptLst>
                <pt modelId="0" type="doc"/>
                <pt modelId="1">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="2">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="3">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="4">
                      <prSet phldr="1"/>
                </pt>
                <pt modelId="5">
                      <prSet phldr="1"/>
                </pt>
          </ptLst>
          <cxnLst>
                <cxn modelId="6" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="7" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
                <cxn modelId="8" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
                <cxn modelId="9" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
                <cxn modelId="10" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
          </cxnLst>
          <bg/>
          <whole/>
    </dataModel>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
</sampData>
```

In this example we define the sample data to consist of five nodes all attached to a document point type node. When displayed, this diagram shows five shapes in the diagram. *end example*]

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SampleData) is located in §A.5.3. *end note*]

21.4.2.27 shape (Shape)

The shape displayed by the containing layout node. Not all layout nodes display shapes.

Attributes	Description
blip (Relationship to Image Part) Namespace: http://purl.oclc.org/ooxml/officeDocument/relationships	Specifies the relationship ID of the explicit relationship to an image which shall be used as the image for the contents of this shape. This relationship shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/image or the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_RelationshipId simple type (§22.8.2.1).
blipPhldr (Image Placeholder)	Specifies whether to use an image placeholder or not. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
hideGeom (Hide Geometry)	When set to "true", hides the geometry of the shape. The text is still visible. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
lkTxEntry (Prevent Text Editing)	Prevents text editing on this shape. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
rot (Rotation)	Rotates the shape by the specified number of degrees. The possible values for this attribute are defined by the W3C XML Schema double datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
type (Shape Type)	Specifies the type of shape. The possible values for this attribute are defined by the ST_LayoutShapeType simple type (§21.4.7.41).
zOrderOff (Z-Order Offset)	Offsets the shape from its default z-order stacking, which is based on the order the layout nodes appear in the XML. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape) is located in §A.5.3. *end note*]

21.4.2.28 style (Shape Style)

This element specifies the style information for a shape, as defined by its DrawingML child elements.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeStyle) is located in §A.4.1. *end note*]

21.4.2.29 styleData (Style Data)

This element defines the style data model which is used to pre-populate a diagram with placeholder data in order for the diagram to display itself in the user interface which shows a quick style applied to the diagram.

[*Example*: Consider the following example of a styleData being used within a DrawingML diagram:

```
<styleData>
    <dataModel>
          <ptLst>
                <pt modelId="0" type="doc"/>
                <pt modelId="1"/>
                <pt modelId="2"/>
          </ptLst>
          <cxnLst>
                <cxn modelId="3" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
                <cxn modelId="4" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
          </cxnLst>
          <bg/>
          <whole/>
    </dataModel>
</styleData>
```

In this example we define a data model which has only two nodes which are shown in the user interface when a layout definition is combined with this data model. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
useDef (Use Default)	If the value of this attribute is `true`, the data model defined in the clrData element is ignored and a default data model is used instead. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SampleData) is located in §A.5.3. *end note*]

21.4.2.30 title (Title)

Title of the Diagram Layout.

Attributes	Description
lang (Language)	Specifies the language of the title or description of this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Value)	Specifies the title or description of this layout definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Name) is located in §A.5.3. *end note*]

21.4.2.31 varLst (Variable List)

This element consists of a list of variables which interact with user interface components.

[*Example*: Consider the following example of a varLst in a DrawingML diagram:

```
<varLst>
    <chMax val="2" />
    <dir val="norm" />
    <resizeHandles val="exact" />
</varLst>
```

In this example we see different variables being defined which modify the behavior of user interface components directly. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutVariablePropertySet) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.3 Data

This section specifies the data that is to be contained within a diagram.

21.4.3.1 bg (Background Formatting)

This element defines formatting that can be applied to the background shape of the entire diagram. The background shape can hold formatting options just as a normal shape can hold within DrawingML.

[*Example*: Consider the following example of a bg in DiagramML:

```
<bg>
     <solidFill>
          <schemeClr val="tx1"/>
     </solidFill>
     <effectLst>
          <glow rad="152400">
               <schemeClr val="accent1">
                    <alpha val="75000"/>
               </schemeClr>
          </glow>
     </effectLst>
</bg>
```

In this example we see a solid fill applied to the background of the diagram along with a glow. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BackgroundFormatting) is located in §A.4.1. *end note*]

21.4.3.2 cxn (Connection)

This element defines a connection between two points. A connection defines a relationship between two points in a diagram.

[*Example*: Consider the following example of a cxn in DiagramML:

```
<cxnLst>
    <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
    <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
    <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
    <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
    <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
    <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
</cxnLst>
```

In this example we see 6 cxn elements defined within a cxnLst element (§21.4.3.3). In this example, a relationship is being defined between point 0 and every other point in the diagram. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
destId (Destination Identifier)	The model identifier of the destination point for a connection. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the destination identifier referencing a point who's model identifier is 4. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
destOrd (Destination Position)	The relative position of the destination point among it's siblings. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the destination position is 0. This means that it is ranked first among its siblings if there are sibling points present. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
modelId (Model Identifier)	The unique identifier associated with this cxn. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the model identifier is 10. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
parTransId (Parent Transition Identifier)	The model identifier of the point representing the parent transition. [*Example*: One example of a parent transition can be thought of as the shape connecting two points, such as an arrow in the diagram. *end example*] The unique identifier associated with this cxn. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0" parTransId="9" sibTransId="5"/>` In this example we see the parent transition identifier is referencing a point who's model identifier is 9. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
presId (Presentation Identifier)	The unique identifier of the layout associated to the cxn (only the active presentation (layout) is saved so all the presId's in the file should be the same). [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" type="presParOf" srcId="0" destId="4" srcOrd="3" destOrd="0" presId="urn:sampleLayouts/layout1"/>` In this example we see the presentation identifier is `urn:sampleLayouts/layout1`. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.
sibTransId (Sibling Transition Identifier)	The model identifier of the point representing the sibling transition. [*Example*: An example of a sibling transition can be thought of as the shape connecting two points, such as an arrow in the diagram. *end example*] [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0" parTransId="9" sibTransId="5"/>` In this example we see the sibling transition identifier is referencing a point who's model identifier is 5. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
srcId (Source Identifier)	The model identifier of the source point for a connection. [*Example*: Consider the following example cxn within DiagramML: `<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>` In this example we see the souce identifier referencing a point who's model identifier is `0`. *end example*] The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
srcOrd (Source Position)	The relative position of the source point among it's siblings. [*Example*: Consider the following example cxn within DiagramML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/> In this example we see the source position is 3. This means that it is ranked third among its siblings. *end example*] The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Point Type)	The type of point, which corresponds to a connection in this case. [*Example*: Consider the following example cxn within DiagramML: <cxn modelId="10" type="presParOf" srcId="0" destId="4" srcOrd="3" destOrd="0" presId="urn:sampleLayouts/layout1"/> In this example we see the point type is defined as presParOf. *end example*] The possible values for this attribute are defined by the ST_CxnType simple type (§21.4.7.23).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Cxn) is located in §A.5.3. *end note*]

21.4.3.3 cxnLst (Connection List)

This element defines a group of connections. There can be a connection list defined for any data model which holds all of the connections between points defined in the diagram.

[*Example*: Consider the following example of a cxnLst in DiagramML:

```
<cxnLst>
    <cxn modelId="7" srcId="0" destId="1" srcOrd="0" destOrd="0"/>
    <cxn modelId="8" srcId="0" destId="2" srcOrd="1" destOrd="0"/>
    <cxn modelId="9" srcId="0" destId="3" srcOrd="2" destOrd="0"/>
    <cxn modelId="10" srcId="0" destId="4" srcOrd="3" destOrd="0"/>
    <cxn modelId="11" srcId="0" destId="5" srcOrd="4" destOrd="0"/>
    <cxn modelId="12" srcId="0" destId="6" srcOrd="5" destOrd="0"/>
</cxnLst>
```

In this example we see 6 cxn elements (§21.4.3.2) defined within a cxnLst element. In this example, a relationship is being defined between point 0 and every other point in the diagram. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CxnList) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.3.4 prSet (Property Set)

This element holds properties and customizations which are used throughout certain elements in DiagramML. The properties can be grouped into the following general categories:

- Presentation Properties - presLayoutVars, style, presAssocId, presName, presStyleLbl, presStyleIdx, presStyleCnt
- Document Properties - loTypeId, loCatId, qsTypeId, qaCatId, csTypeId, coherent3DOff
- Semantic Element Properties - phldrT, phldr
- Customization Properties - custAng, custFlipVert, custFlipHor, custSzX, custSzY, custScaleX, custScaleY, custT, custLinFactX, custLinFactY, custLinFactNeighborX, custLinFactNeighborY, custRadScaleRad, custRadScaleInc

[*Example*: Consider the basic example of prSet in use in DrawingML on a document point type:

```
<prSet loTypeId="urn:microsoft.com/office/officeart/2005/8/layout/default"
loCatId="list"
qsTypeId="urn:microsoft.com/office/officeart/2005/8/quickstyle/3d1" qsCatId="3D"
csTypeId="urn:microsoft.com/office/officeart/2005/8/colors/colorful2"
csCatId="colorful" phldr="1"/>
```

In this example we define the layout identifier, the category of the layout, the quick style identifier, the quick style category, along with the color style and color style category. *end example*]

Attributes	Description
coherent3DOff (Coherent 3D Behavior)	Enables or disables the Coherent 3D behavior for styles that specify this property. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
csCatId (Color Transform Category)	This attribute specifies the identifier of the current color transform category. The possible values for this attribute are defined by the W3C XML Schema string datatype.
csTypeId (Color Transform Type Identifier)	This attribute specifies the identifier of the currently applied color transform. The possible values for this attribute are defined by the W3C XML Schema string datatype.
custAng (Custom Rotation)	Specifies the amount that rotation is customized by. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custFlipHor (Custom Horizontal Flip)	Specifies if there is a custom horizontal flip applied. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
custFlipVert	Specifies if there is a custom vertical flip applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
(Custom Vertical Flip)	The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
custLinFactNeighborX (Neighbor Offset Width)	Specifies the percentage of the neighbor's width used for offsetting shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactNeighborY (Neighbor Offset Height)	Specifies the percentage of the neighbor's height used for offsetting shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactX (Custom Factor Width)	Specifies the percentage of the current shape width used for offsetting the shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custLinFactY (Custom Factor Height)	Specifies the percentage of the current shape height used for offsetting the shape. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custRadScaleInc (Include Angle Scale)	Specifies the amount that the include angle has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custRadScaleRad (Radius Scale)	Specifies how much the radius has been scaled. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custScaleX (Width Scale)	Specifies the amount that the width has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custScaleY (Height Scale)	Specifies the amount that the height has been scaled by. The possible values for this attribute are defined by ST_PrSetCustVal simple type (§21.4.7.66).
custSzX (Fixed Width Override)	Specifies a fixed width override for a shape. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custSzY (Fixed Height Override)	Specifies a fixed height override for a shape. The possible values for this attribute are defined by the W3C XML Schema int datatype.
custT (Text	Specifies if the text has been customized which allows layout to ignore automatic

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Changed)	formatting options available to the text. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
loCatId (Current Diagram Category)	Specifies the current identifier of the layout category applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
loTypeId (Current Diagram Type)	Specifies the identifier for the layout currently applied to the diagram. The possible values for this attribute are defined by the W3C XML Schema string datatype.
phldr (Placeholder)	Indicates that the point is a placeholder or sample item. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.
phldrT (Placeholder Text)	The text used for display in the element if the placeholder flag is set to `true`. If this property is not set then the default placeholder text is used. The possible values for this attribute are defined by the W3C XML Schema string datatype.
presAssocID (Presentation Element Identifier)	The point associated with this presentation element. This identifier is used together with presName to create a unique key for presentation element indexing. The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).
presName (Presentation Name)	The layout node name of this presentation element. This name is used together with presAssocID to create a unique key for presentation element indexing. The possible values for this attribute are defined by the W3C XML Schema string datatype.
presStyleCnt (Presentation Style Count)	Specifies the layout node style count of this presentation element. The possible values for this attribute are defined by the W3C XML Schema int datatype.
presStyleIdx (Presentation Style Index)	Specifies the layout node style index of this presentation element. The possible values for this attribute are defined by the W3C XML Schema int datatype.
presStyleLbl (Presentation Style Label)	Specifies the layout node style label of this presentation element. The possible values for this attribute are defined by the W3C XML Schema string datatype.
qsCatId (Current	Specifies the identifier of the category of the currently applied quick style.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
Style Category)	The possible values for this attribute are defined by the W3C XML Schema string datatype.
qsTypeId (Current Style Type)	Specifies the identifier of the currently applied quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ElemPropSet) is located in §A.5.3. *end note*]

21.4.3.5 pt (Point)

This element defines a point in DiagramML. A point in DiagramML is defined to hold data associated with a particular point or node in a diagram. Transitions between nodes in a diagram along with the nodes themselves are defined as different types of points. A point is not only responsible for holding the data associated with a node in a diagram, but also for holding customization properties made to the text and shape associated with the particular node.

[*Example*: Consider the following example of a pt in DiagramML:

```
<pt modelId="{C6A8900D-3F1B-4F1D-A514-4E8BDD964568}">
    <prSet phldrT="[Text]"/>
    <spPr/>
    <t>
          <bodyPr/>
          <lstStyle/>
          <p>
                <r>
                      <rPr lang="en-US" smtClean="0"/>
                      <t>Text 2</t>
                </r>
                <endParaRPr lang="en-US" dirty="0"/>
          </p>
    </t>
</pt>
```

In this example we define a point which holds the data associated with a node in a diagram. The actual text in the diagram is defined in the text body, t, tag and consists of the string "Text 2". There are no overrides made to the shape properties and placeholder text defined for this node when there is no text body present. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

cxnId (Connection Identifier)	The model identifier of the connection that represents the transition node.
	
	[*Example*: Consider the following example of a `cxnId`:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example we define the connection related to this point to reference connection 9. *end example*]
	
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).

modelId (Model Identifier)	The unique identifier of the element within the data model. This identifier should be unique only relative to the containing data model.
	
	[*Example*: Consider the following example of a `cxnId`:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example we define the point type is to be 5. *end example*]
	
	The possible values for this attribute are defined by the ST_ModelId simple type (§21.4.7.43).

type (Point Type)	The type of point.
	
	[*Example*: Consider the following example of a `cxnId`:
	
	`<dgm:pt modelId="5" type="parTrans" cxnId="9">`
	`  <dgm:prSet/>`
	`  <dgm:spPr/>`
	`</dgm:pt>`
	
	In this example the point type is defined as a `parTrans` point type. *end example*]
	
	The possible values for this attribute are defined by the ST_PtType simple type (§21.4.7.51).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Pt) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.4.3.6 ptLst (Point List)

This element simply holds a list of points within the data model.

[*Example*: Consider the following example of a very simple point list in DiagramML:

```
<dgm:ptLst>
    <dgm:pt modelId="0" type="doc"/>
    <dgm:pt modelId="1"/>
    <dgm:pt modelId="2"/>
    <dgm:pt modelId="3"/>
    <dgm:pt modelId="4"/>
    <dgm:pt modelId="5"/>
    <dgm:pt modelId="6"/>
</dgm:ptLst>
```

In this example we define a single document type point and five node type points. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PtList) is located in §A.5.3. *end note*]

21.4.3.7 spPr (Shape Properties)

This element specifies the properties for a single shape in a diagram's data, as defined using DrawingML child elements.

Attributes	Description
bwMode (Black and White Mode) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Specifies that the picture should be rendered using only black and white coloring. That is the coloring information for the picture should be converted to either black or white when rendering the picture. No gray is to be used in rendering this image, only stark black and stark white. [*Note*: This does not mean that the picture itself that is stored within the file is necessarily a black and white picture. This attribute instead sets the rendering mode that the picture has applied to when rendering. *end note*] The possible values for this attribute are defined by the ST_BlackWhiteMode simple type (§20.1.10.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ShapeProperties) is located in §A.4.1. *end note*]

21.4.3.8 t (Text Body)

Text body containing the default body, paragraph and character properties. There should be a signle paragraph and no text runs. Any runs in the first paragraph and paragraphs in addition to the first are ignored.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextBody) is located in §A.4.1. *end note*]

21.4.3.9 whole (Whole E2O Formatting)

Formatting that applies to the entire diagram object, and not just the background, includes line and effect properties.

[*Note*: The W3C XML Schema definition of this element's content model (CT_WholeE2oFormatting) is located in §A.4.1. *end note*]

21.4.4 Color Information

This section defines the coloring information that is to be associated with a diagram.

21.4.4.1 cat (Color Transform Category)

This element specifies the category in the user interface that a color transform is to be displayed within.

[*Example*: Consider the following example of a cat in DiagramML:

```
<dgm:cat type="mainScheme" pri="10300"/>
```

In this example we see a cat defined with a category type and priority. *end example*]

Attributes	Description
pri (Priority)	The priority within the category for this color variation determines the order in which it displays in the user interface. The lower numbers are to be displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	The category type used to organize the color transforms in the user interface. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTCategory) is located in §A.5.3. *end note*]

21.4.4.2 catLst (Color Transform Category List)

This element defines a list of color transform categories. This list can be used to populate user interface components which could separate color transforms into categories.

[*Example*: Consider the following example of a catLst in DiagramML:

```
<dgm:catLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <dgm:cat type="mainScheme" pri="10300"/>
    </dgm:catLst>
```

In this example we see a catLst defined which holds a single *color transform category* (§21.4.4.1). *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTCategories) is located in §A.5.3. *end note*]

21.4.4.3 colorsDef (Color Transform Definitions)

This element is the root element for color transforms. Held within this element are all of the available color transforms themselves along with other elements and attributes associated with defining the general color transform properties and attributes.

[*Example*: Consider the following example of a colorsDef in DiagramML:

```
<dgm:colorsDef
xmlns:dgm="http://schemas.openxmlformats.org/drawingml/2006/diagram"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
uniqueId="urn:microsoft.com/office/officeart/2005/8/colors/accent0_3"
minVer="12.0">
    <dgm:title lang="" val=""/>
    <dgm:desc lang="" val=""/>
    <dgm:catLst>
          <dgm:cat type="mainScheme" pri="10300"/>
    </dgm:catLst>
    <dgm:styleLbl name="node0">
          …
    </dgm:styleLbl>
    …
</dgm:colorsDef>
```

In this example we see a sampling of a colorsDef being defined with a number of styleLbl elements held within the colorsDef. *end example*]

Attributes	Description

minVer (Minimum Version)	The minimum product version that can support this color transform.
	
	[*Example*: Consider the following example:
	
	`<colorsDef uniqueId="urn:colors/accent0_3" minVer="12.0">`
	`…`
	`</colorsDef>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	In this example we see the minVer set to `12.0`. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

uniqueId (Unique ID)	A unique id associated with the color transform definition.
	
	[*Example*: Consider the following example:
	
	`<colorsDef uniqueId="urn:colors/accent0_3" minVer="12.0">`
	`…`
	`</colorsDef>`
	
	In this example we see the uniqueId set to `urn:colors/accent0_3`. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorTransform) is located in §A.5.3. *end note*]

21.4.4.4 colorsDefHdr (Color Transform Definition Header)

This element specifies header information associated with a color transform definition. The header information is used by an application to preprocess required data in order to help with possible performance concerns associated with an initial full load of a color transform definition.

[*Example*: Consider the following example of a colorsDefHdr within DiagramML:

```
<colorsDefHdr uniqueId="urn:colors/accent0_1">
    <title val="Main 1" />
    <desc val="" />
    <catLst>
        <cat type="mainScheme" pri="10100" />
    </catLst>
</colorsDefHdr>
```

In this example we see a color transform definition header which defines a title and category for a set of color transforms. *end example*]

Attributes	Description
minVer (Minimum Version)	The minimum product version that can support the associated color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
resId (Resource ID)	This attribute is the id which associates this header to the actual color transform definition. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique ID)	This attribute defines a unique identifier for the associated color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorTransformHeader) is located in §A.5.3. *end note*]

21.4.4.5 colorsDefHdrLst (Color Transform Header List)

This element is simply a list of color transform definition headers and is used to consolidate multiple headers in a group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_ColorTransformHeaderLst) is located in §A.5.3. *end note*]

21.4.4.6 desc (Description)

This element holds a description for a color definition. The description can be used to describe the qualities associated with a particular color transform definition.

Attributes	Description
lang (Language)	The natural language of the color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	The string which is used as the description of the color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTDescription) is located in §A.5.3. *end note*]

21.4.4.7 effectClrLst (Effect Color List)

This element defines a list of colors applied to effects within a color transform.

[*Example*: Consider the following example of an effectClrLst in DiagramML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<dgm:effectClrLst meth="repeat">
    <a:schemeClr val="dk2">
          <a:tint val="60000"/>
    </a:schemeClr>
</dgm:effectClrLst>
```

In this example we see a single color defined in the effectClrLst, more specifically we see a scheme color being utilized with a tint applied to the color. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

21.4.4.8 fillClrLst (Fill Color List)

This element defines a list of colors which are used as fill colors in the color transform. The fill colors define the color of the nodes in a diagram.

[*Example*: Consider the following example of a fillClrLst in DiagramML:

```
<dgm:fillClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:fillClrLst>
```

In this example the fillClrList contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.9 linClrLst (Line Color List)

This element defines a list of colors which are used as line colors in the color transform. The line colors define the color of the lines used on a given node in a diagram

[*Example*: Consider the following example of a lnClrLst in DiagramML:

```
<dgm:lnClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:lnClrLst>
```

In this example the lnClrList contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*]
	
	The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the `repeat` color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	(§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.10 styleLbl (Style Label)

This element defines a style label. The style label is used to define a color transform that is applied to a given node in a diagram.

[*Example*: Consider the following example of a styleLbl in DiagramML:

```
<dgm:styleLbl name="exampleStyleLabel">
    <dgm:fillClrLst>
          <a:schemeClr val="accent2"/>
          <a:schemeClr val="accent3"/>
    </dgm:fillClrLst>
    <dgm:linClrLst meth="repeat">
          <a:schemeClr val="lt1"/>
    </dgm:linClrLst>
    <dgm:effectClrLst/>
    <dgm:txLinClrLst/>
    <dgm:txFillClrLst/>
    <dgm:txEffectClrLst/>
</dgm:styleLbl>
```

In this example we see a style label defined in its entirety. This style label can be used on a layout node in order to define the color transform that is to be applied to the node. *end example*]

Attributes	Description

name (Name)	A name given to the style label. This name can be referenced by layout nodes in order to apply the style label to the layout node.
	
	[*Example*: Consider the following example of a styleLbl in DiagramML:
	`<dgm:styleLbl name="exampleStyleLabel">`
	…
	`</dgm:styleLbl>`
	
	In this example we see a style label defined with the name `exampleStyleLabel` defined. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTStyleLabel) is located in §A.5.3. *end note*]

21.4.4.11 title (Title)

The name or title given to the color definition header.

Attributes	Description
lang (Language)	The natural language of the title or description of a color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	A string used for a description of a color transform definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CTName) is located in §A.5.3. *end note*]

21.4.4.12 txEffectClrLst (Text Effect Color List)

This element defines a list of colors which are used as text effect colors in the color transform. The text effect colors define the color of the text effects used on a given node in a diagram

[*Example*: Consider the following example of a txEffectClrLst in DiagramML:

```
<dgm:txEffectClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:txEffectClrLst>
```

In this example the txEffectClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`  <a:schemeClr val="dk2">`
	`  <a:tint val="60000"/>`
	`  </a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example an lnClrLst is defined with a hue direction defined as clockwise. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	example] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).

meth (Color Application Method Type)	The method used to apply the color transform.
	
	[*Example*: Consider the following example of a meth in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`<a:schemeClr val="dk2">`
	`<a:tint val="60000"/>`
	`</a:schemeClr>`
	`</dgm:lnClrLst>`
	
	In this example and lnClrLst is defined using the repeat color application method. *end example*]
	
	The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.13 txFillClrLst (Text Fill Color List)

This element defines a list of colors which are used as text colors in the color transform. The text colors define the color of the text used in a given node in a diagram

[*Example*: Consider the following example of a txFillClrLst in DiagramML:

```
<dgm:txFillClrLst meth="repeat">
     <a:schemeClr val="dk2"/>
</dgm:txFillClrLst>
```

In this example the txFillClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description

hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs.
	
	[*Example*: Consider the following example of a hueDir in use:
	
	`<dgm:lnClrLst hueDir="cw" meth="repeat">`
	`<a:schemeClr val="dk2">`
	`<a:tint val="60000"/>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</a:schemeClr> </dgm:lnClrLst> In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).
meth (Color Application Method Type)	The method used to apply the color transform. [*Example*: Consider the following example of a meth in use: <dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example and lnClrLst is defined using the repeat color application method. *end example*] The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.4.14 txLinClrLst (Text Line Color List)

This element defines a list of colors which are used as text line colors in the color transform. The text line colors define the color of the line on text used in a given node in a diagram

[*Example*: Consider the following example of a txLinClrLst in DiagramML:

```
<dgm:txLinClrLst meth="repeat">
    <a:schemeClr val="dk2"/>
</dgm:txLinClrLst>
```

In this example the txLinClrLst contains a single scheme color and is utilizing the repeat method for color application. *end example*]

Attributes	Description
hueDir (Hue Direction)	The direction around the color wheel the hue shift (if defined) occurs. [*Example*: Consider the following example of a hueDir in use:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example an lnClrLst is defined with a hue direction defined as clockwise. *end example*] The possible values for this attribute are defined by the ST_HueDir simple type (§21.4.7.38).
meth (Color Application Method Type)	The method used to apply the color transform. [*Example*: Consider the following example of a meth in use: <dgm:lnClrLst hueDir="cw" meth="repeat"> <a:schemeClr val="dk2"> <a:tint val="60000"/> </a:schemeClr> </dgm:lnClrLst> In this example and lnClrLst is defined using the repeat color application method. *end example*] The possible values for this attribute are defined by the ST_ClrAppMethod simple type (§21.4.7.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Colors) is located in §A.5.3. *end note*]

21.4.5 Style Definitions

This section describes the styling information to be associated with a diagram.

21.4.5.1 cat (Category)

The category in the user interface where this quick style displays in the user interface.

[*Example*: Consider the following example of a cat in use in DiagramML:

```
<catLst>
    <cat type="3D" pri="11100"/>
</catLst>
```

In this example we see a 3D category type being defined with a priority of 11100. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
pri (Priority)	The priority within the category for this style determines the order in which it displays in the user interface. Lower numbers are displayed at the beginning of the list. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.
type (Category Type)	Category type. This is used to organize the quick style in the user interface. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDCategory) is located in §A.5.3. *end note*]

21.4.5.2 catLst (Category List)

This element is simply a list of categories.

[*Example*: Consider the following example of a catLst in use in DiagramML:

```
<catLst>
    <cat type="Simple" pri="10000"/>
    <cat type="3D" pri="11100"/>
</catLst>
```

In this example two categories defined in the category list. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDCategories) is located in §A.5.3. *end note*]

21.4.5.3 desc (Style Label Description)

This element defines a description for a style label definition. The description is simply a string describing the characteristics of the style label definition.

[*Example*: Consider the following example of the desc element in DiagramML:

```
<desc lang="" val="3-D Style 1"/>
```

In this example we define the description to be `3-D Style 1`. *end example*]

Attributes	Description
lang (Natural Language)	The natural language of the title or description of this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Description Value)	The string used for the description. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDDescription) is located in §A.5.3. *end note*]

21.4.5.4 presLayoutVars (Presentation Layout Variables)

This element specified the layout property set. This set of properties determine different aspects concerning the layout of a diagram. All of the elements associated with enabling or disabling aspects of the user interface are also defined here.

[*Note*: The W3C XML Schema definition of this element's content model (CT_LayoutVariablePropertySet) is located in §A.5.3. *end note*]

21.4.5.5 scene3d (3-D Scene)

The 3-D scene which consists of a camera, a light rig, and an optional backdrop to catch shadows.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Scene3D) is located in §A.4.1. *end note*]

21.4.5.6 sp3d (3-D Shape Properties)

A set of 3-D properties which a shape can contain.

Attributes	Description
contourW (Contour Width) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the width of the contour on the shape. [*Example*: Consider the following example of a contourW in use within the sp3d element: `<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"><a:bevelT w="254000" h="254000"/><a:bevelB w="254000" h="254000"/><a:extrusionClr><a:srgbClr val="FF0000"/></a:extrusionClr><a:contourClr><a:schemeClr val="accent3"/></a:contourClr></a:sp3d>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	In this example, we see a countourW defined as `50800`. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
extrusionH (Extrusion Height) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the height of the extrusion applied to the shape. [*Example*: Consider the following example of an extrusionH in use within the sp3d element: `<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"><a:bevelT w="254000" h="254000"/><a:bevelB w="254000" h="254000"/><a:extrusionClr><a:srgbClr val="FF0000"/></a:extrusionClr><a:contourClr><a:schemeClr val="accent3"/></a:contourClr></a:sp3d>` In this example, we see a extrusionH defined as `165100`. *end example*] The possible values for this attribute are defined by the ST_PositiveCoordinate simple type (§20.1.10.41).
prstMaterial (Preset Material Type) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the preset material which is combined with the lighting properties to give the final look and feel of a shape. [*Example*: Consider the following example of a prstMaterial in use within the sp3d element: `<a:sp3d extrusionH="165100" contourW="50800" prstMaterial="plastic"><a:bevelT w="254000" h="254000"/><a:bevelB w="254000" h="254000"/><a:extrusionClr><a:srgbClr val="FF0000"/></a:extrusionClr><a:contourClr><a:schemeClr val="accent3"/></a:contourClr></a:sp3d>` In this example, we see a prstMaterial defined as `plastic`. *end example*] The possible values for this attribute are defined by the ST_PresetMaterialType simple

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	type (§20.1.10.50).
z (Shape Depth) Namespace: http://purl.oclc.org/ooxml/drawingml/main	Defines the z coordinate for the 3D shape. The possible values for this attribute are defined by the ST_Coordinate simple type (§20.1.10.16).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shape3D) is located in §A.4.1. *end note*]

21.4.5.7 styleDef (Style Definition)

This element is the root tag for a style definition.

[*Example*: Consider the following example of a styleDef in DiagramML:

```
<dgm:styleDef
xmlns:dgm="http://schemas.openxmlformats.org/drawingml/2006/diagram"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
uniqueId="urn:microsoft.com/office/officeart/2005/8/quickstyle/3d1"
minVer="12.0">
    <dgm:title lang="" val="3-D Style 1"/>
    <dgm:desc lang="" val="3-D Style 1"/>
    <dgm:catLst>
          <dgm:cat type="3D" pri="11100"/>
    </dgm:catLst>
    <dgm:scene3d>
          <a:camera prst="orthographicFront"/>
          <a:lightRig rig="threePt" dir="t"/>
    </dgm:scene3d>
    <dgm:style>
                  …
    </dgm:style>
    <dgm:styleLbl name="node0">
          <dgm:scene3d>
                <a:camera prst="orthographicFront"/>
                <a:lightRig rig="flat" dir="t"/>
          </dgm:scene3d>
          <dgm:sp3d prstMaterial="flat">
                <a:bevelT w="120900" h="88900"/>
                <a:bevelB w="88900" h="31750" prst="angle"/>
          </dgm:sp3d>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <dgm:txPr/>
            <dgm:style>
                  <a:lnRef idx="0">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:lnRef>
                  <a:fillRef idx="3">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:fillRef>
                  <a:effectRef idx="2">
                        <a:scrgbClr r="0" g="0" b="0"/>
                  </a:effectRef>
                  <a:fontRef idx="minor">
                        <a:schemeClr val="lt1"/>
                  </a:fontRef>
            </dgm:style>
      </dgm:styleLbl>
   …
   </dgm:styleDef>
```

In this example we see a styleDef being defined along with many properties. *end example*]

Attributes	Description

minVer (Minimum Version)	The minimum product version that can support this quick style.
	
	[*Example*: Consider the following example of a styleDef in DiagramML:
	
	`<styleDef uniqueId="urn:quickstyle/3d1" minVer="12.0">`
	
	…
	
	`</styleDef>`
	
	In this example we see the `minVer` defined to be `12.0`. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

uniqueId (Unique Style ID)	Unique ID that identifies a style. [*Example*: Consider the following example of a styleDef in DiagramML: `<styleDef uniqueId="urn:quickstyle/3d1" minVer="12.0">` …

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	</styleDef> In this example we see the uniqueId defined to be urn:quickstyle/3d1. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleDefinition) is located in §A.5.3. *end note*]

21.4.5.8 styleDefHdr (Style Definition Header)

This element specifies header information associated with a style definition. The header information is used by an application to preprocess required data in order to help with possible performance concerns associated with an initial full load of a color transform definition.

[*Example*: Consider the following example of a styleDefHdr element within DiagramML:

```
<styleDefHdr uniqueId="urn:quicktyle/3d1">
    <title val="3D" />
    <desc val="" />
    <catLst>
        <cat type="3D" pri="10100" />
    </catLst>
</ styleDefHdr >
```

In this example we see a style definition header which defines a title and category for a set of style definitions. *end example*]

Attributes	Description
minVer (Minimum Version)	The minimum product version that can support this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.
resId (Resource ID)	This attribute is the id which associates this header to the actual style definition part. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uniqueId (Unique Style ID)	This attribute defines a unique identifier for the associated style definition. The possible values for this attribute are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleDefinitionHeader) is located in §A.5.3. *end note*]

21.4.5.9 styleDefHdrLst (List of Style Definition Headers)

This element is simply a list of style definition headers and is used to consolidate multiple headers into one group.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleDefinitionHeaderLst) is located in §A.5.3. *end note*]

21.4.5.10 styleLbl (Style Label)

This element defines the actual style which is applied to a node in a diagram The style is referenced from within layout node. The style label contains formatting (without defining color) such as the 3D properties and text properties associated with a shape.

[*Example*: Consider the following example of a styleLbl in DiagramML:

```
<styleLbl name="node0">
    <scene3d>
          <camera prst="orthographicFront"/>
          <lightRig rig="flat" dir="t"/>
    </scene3d>
    <sp3d prstMaterial="flat">
          <bevelT w="120900" h="88900"/>
          <bevelB w="88900" h="31750" prst="angle"/>
    </sp3d>
    <txPr/>
    <style>
          <lnRef idx="0">
                <scrgbClr r="0" g="0" b="0"/>
          </lnRef>
          <fillRef idx="3">
                <scrgbClr r="0" g="0" b="0"/>
          </fillRef>
          <effectRef idx="2">
                <scrgbClr r="0" g="0" b="0"/>
          </effectRef>
          <fontRef idx="minor">
                <schemeClr val="lt1"/>
          </fontRef>
    </style>
</styleLbl>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example we see a styleLbl defined which sets 3D properties for the scene, shape 3D properties, line, fill, effect and font properties. *end example*]

Attributes	Description
name (Style Name)	The name of the style. This apears as the tooltip in the user interface. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_StyleLabel) is located in §A.5.3. *end note*]

21.4.5.11 title (Title)

This element defines the title given to a style definition header. The title is simply a name for the style definition.

[*Example*: Consider the following example of title being used in DiagramML:

```
<styleDefHdr uniqueId="urn:quickstyle/3d1" minVer="12.0">
     <title val="My Title"/>
     <desc val="My Description"/>

     …

</styleDefHdr>
```

In this example we see the title being set to `My Title`. *end example*]

Attributes	Description
lang (Natural Language)	The natural language of the title or description of this quick style. The possible values for this attribute are defined by the W3C XML Schema string datatype.
val (Description Value)	The string used for the description. The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_SDName) is located in §A.5.3. *end note*]

21.4.5.12 txPr (Text Properties)

This element defines special text formatting that can be applied to text through a style label.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextProps) is located in §A.5.3. *end note*]

21.4.6 Layout Definition

This section specifies the node layout information to be associated with a diagram.

21.4.6.1 animLvl (Level Animation)

This variable is used to indicate the animate by level string which is displayed to a user in the user interface.

[*Example*: Consider the following example of animLvl in use in DiagramML:

```
<varLst>
     <chMax val="1" />
     <dir val="norm" />
     <animLvl val="ctr" />
     <resizeHandles val="exact" />
</varLst>
```

In this example we see that the animLvl is set to `ctr`. This is being defined in a radial type diagram which allows the user to specify that animation is to start at the center of the diagram. *end example*]

Attributes	Description
val (Level Animation Value)	This attribute indicates the string to use for level animation in the user interface. The possible values for this attribute are defined by the ST_AnimLvlStr simple type (§21.4.7.2).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimLvl) is located in §A.5.3. *end note*]

21.4.6.2 animOne (One by One Animation String)

This variable is used to indicate the string to use for one-by-one animation in the user interface. This is used primarily when defining hierarchical diagrams to specify different ways animations applies to different levels of the diagram.

[*Example*: Consider the following example of animOne used in a hierarchical type diagram:

```
<varLst>
     <chPref val="1" />
     <dir val="norm" />
     <animOne val="branch" />
     <animLvl val="lvl" />
     <resizeHandles val="exact" />
</varLst>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In this example we see that the animOne element is defined to animate the diagram per branch. *end example*]

Attributes	Description
val (One By One Animation Value)	Specifies the type of one-by-one animation to use for a diagram. The possible values for this attribute are defined by the ST_AnimOneStr simple type (§21.4.7.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_AnimOne) is located in §A.5.3. *end note*]

21.4.6.3 bulletEnabled (Show Insert Node)

This element is used to indicate whether to enable user interface components associated with inserting a node in the data model.

[*Example*: Consider the following example of bulletEnabled in DiagramML:

```
<varLst>
    <bulletEnabled val="true" />
</varLst>
```

In this example, an insert button in the user interface is to be enabled when the focus is within the containing layout node. *end example*]

Attributes	Description
val (Show Insert Node Value)	This attribute is used to indicatewhether a user interface for inserting a node should be enabled. A value of `true` indicates that the user interface should be enabled. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_BulletEnabled) is located in §A.5.3. *end note*]

21.4.6.4 chMax (Maximum Children)

This element is used to indicate when to enable and disable the user interface components associated with adding a new shape to a diagram. This element defines a max number of nodes a diagram can support through the user interface directly.

[*Example*: Consider the following example of chMax usage in DiagramML:

```
<varLst>
    <chMax val="5"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <dir val="norm"/>
    <resizeHandles val="exact" />
</varLst>
```

In this example we define the user interface to only be enabled to insert five nodes. *end example*]

Attributes	Description
val (Maximum Children Value)	This attribute indicates the maximum number of children the node can have before the user interface should be disabled. A value of `-1` indicates an infinite number of children. Default value is `-1`. The possible values for this attribute are defined by the ST_NodeCount simple type (§21.4.7.44).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChildMax) is located in §A.5.3. *end note*]

21.4.6.5 chPref (Preferred Number of Children)

This variable indicates the number of children that the current node prefers to have. [*Note*: For example, this could be used to guide how many shapes to add by default to a diagram at various levels in the hierarchy. *end note*]

[*Example*: Consider the following example of chPref being used in DiagramML:

```
<varLst>
    <chMax val="3" />
    <chPref val="1" />
    <dir val="norm" />
    <animLvl val="lvl" />
    <resizeHandles val="rel" />
</varLst>
```

In this example, chPref is set to a single node and the associated user interface may disable insertion of further nodes after a single node has been inserted. *end example*]

Attributes	Description
val (Preferred Number of CHildren Value)	This attribute indicates the number of children that the current node prefers to have. A value of `-1` indicates an infinite number of children. Default value is `-1`. The possible values for this attribute are defined by the ST_NodeCount simple type (§21.4.7.44).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_ChildPref) is located in §A.5.3. *end note*]

21.4.6.6 dir (Diagram Direction)

This element indicates whether the diagram should switch direction. This element provides the ability to define different behavior for diagrams considering LTR or RTL directions.

Attributes	Description
val (Diagram Direction Value)	This variable indicates whether the diagram should switch direction. The possible values for this attribute are defined by the ST_Direction simple type (§21.4.7.26).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Direction) is located in §A.5.3. *end note*]

21.4.6.7 hierBranch (Organization Chart Branch Style)

This element defines the layout style of a branch in an organizational chart.

[*Example*: Consider the following example of hierBranch being used in DiagramML:

```
<varLst>
    <hierBranch val="init" />
</varLst>
```

In this example the value of hierBranch is defined as `init` which is a kind of not set state, or initial state. *end example*]

Attributes	Description
val (Organization Chart Branch Style Value)	The value of this attribute indicates the layout style of a branch in an organization chart. The default value is `std`. The possible values for this attribute are defined by the ST_HierBranchStyle simple type (§21.4.7.37).

[*Note*: The W3C XML Schema definition of this element's content model (CT_HierBranchStyle) is located in §A.5.3. *end note*]

21.4.6.8 orgChart (Show Organization Chart User Interface)

This element is used to indicate when to show user interface controls specifically associated with organizational charts such as being able to add an assistant to a selected node.

[*Example*: Consider the following example of orgChart used in DiagramML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<varLst>
    <orgChart val="true" />
    <chPref val="1" />
    <dir val="norm" />
    <animOne val="branch" />
    <animLvl val="lvl" />
    <resizeHandles val="rel" />
</varLst>
```

In this example we set the orgChart value to `true` indicating that the organizational chart specific user interface controls are to be enabled when the containing diagram is used. *end example*]

Attributes	Description
val (Show Organization Chart User Interface Value)	This attribute value specifies when to show the 'Insert Assistant' user interface control and the 'Change Layout' user interface for this diagram. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_OrgChart) is located in §A.5.3. *end note*]

21.4.7 Simple Types

This is the complete list of simple types dedicated to DrawingML – Diagrams.

21.4.7.1 ST_AlgorithmType (Algorithm Types)

Types of available algorithms.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
composite (Composite)	The composite algorithm specifies the size and position for all child layout nodes. You can use it to create graphics with a predetermined layout or in combination with other algorithms to create more complex shapes.
conn (Connector Algorithm)	The connector algorithm lays out and routes connecting lines, arrows, and shapes between layout nodes.
cycle (Cycle Algorithm)	The cycle algorithm lays out child layout nodes around a circle or portion of a circle using equal angle spacing.
hierChild (Hierarchy Child Algorithm)	The hierarchy child algorithm works with the hierRoot

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	algorithm to create hierarchical tree layouts. This algorithm aligns and positions its child layout nodes in a linear path under the hierRoot layout node.
hierRoot (Hierarchy Root Algorithm)	The hierarchy root algorithm works with the hierChild algorithm to create hierarchical tree layouts. The hierRoot algorithm aligns and positions the hierRoot layout node in relation to the hierChild layout nodes.
lin (Linear Algorithm)	The linear algorithm lays out child layout nodes along a linear path.
pyra (Pyramid Algorithm)	The pyramid algorithm lays out child layout nodes along a vertical path and works with the trapezoid shape to create a pyramid.
snake (Snake Algorithm)	The snake algorithm lays out child layout nodes along a linear path in two dimensions, allowing the linear flow to continue across multiple rows or columns.
sp (Space Algorithm)	The space algorithm is used to specify a minimum space between other layout nodes or as an indication to do nothing with the layout node's size and position.
tx (Text Algorithm)	The text algorithm sizes text to fit inside a shape and controls its margins and alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AlgorithmType) is located in §A.5.3. *end note*]

21.4.7.2 ST_AnimLvlStr (Animation Level String Definition)

This simple type specifies the possible values for the string that should be displayed by a consumer for level animation of this diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (From Center Animation)	This value specifies that the consumer shall allow "From Center At Once" or "From Center One by One" animation styles for this diagram.
lvl (By Level Animation)	This value specifies that the consumer shall display "By Level" animation types for this diagram.
none (Disable Level At Once)	This value specifies that the consumer shall disable level at once animation.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimLvlStr) is located in §A.5.3. *end note*]

21.4.7.3 ST_AnimOneStr (One by One Animation Value Definition)

This simple type defines the possible values for the string to use for one by one animation in the UI. Default value is one.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
branch (By Branch One By One)	This value specifies that the one by one animation string in the user interface should read "By Branch One By One".
none (Disable One-by-One)	This value specifies that the consumer should disable one by one animation.
one (One By One)	This value specifies that the one by one animation string in the user interface should read "One By One".

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AnimOneStr) is located in §A.5.3. *end note*]

21.4.7.4 ST_ArrowheadStyle (Arrowhead Styles)

This simple type defines different arrowhead style types for connectors.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
arr (Arrowhead Present)	Specifies that an arrowhead is to be used on the connector.
auto (Auto)	Specifies that the algorithm defines if an arrowhead is to be used on a connector.
noArr (No Arrowhead)	Specifies no arrowhead is to be used on the connector.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ArrowheadStyle) is located in §A.5.3. *end note*]

21.4.7.5 ST_AutoTextRotation (Auto Text Rotation)

This simple type defines how text rotates within a shape when the shape is rotated by an algorithm during layout.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
grav (Gravity)	Specifies that when the angle of the text hits the threshold of 90 degrees and 180 degrees, the text rotates by 180 degrees.
none (None)	Specifies that text always rotates with the shape.
upr (Upright)	Specifies that when the text angle hits 45, 135, 225, or 315 degree thresholds, then it rotates by negative 90 degrees.

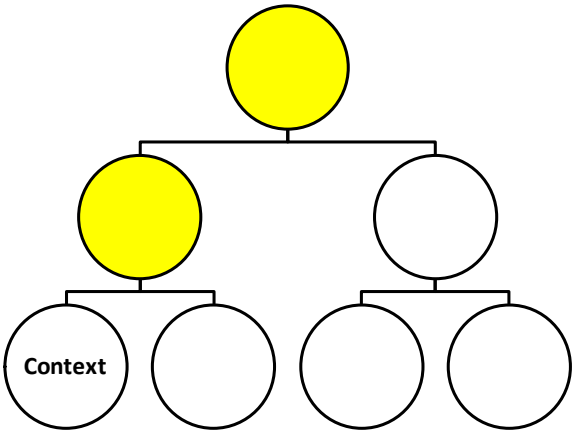
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AutoTextRotation) is located in §A.5.3. *end note*]

21.4.7.6 ST_AxisType (Axis Type)

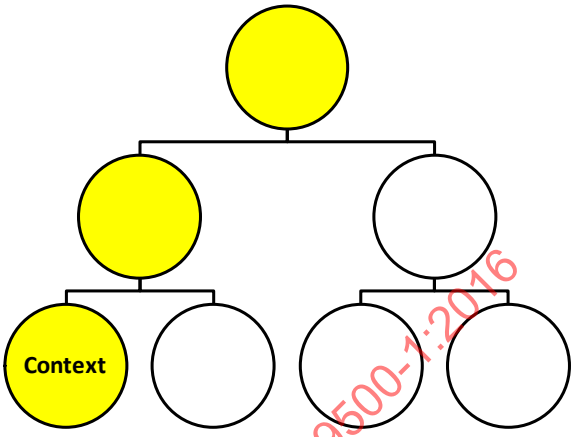
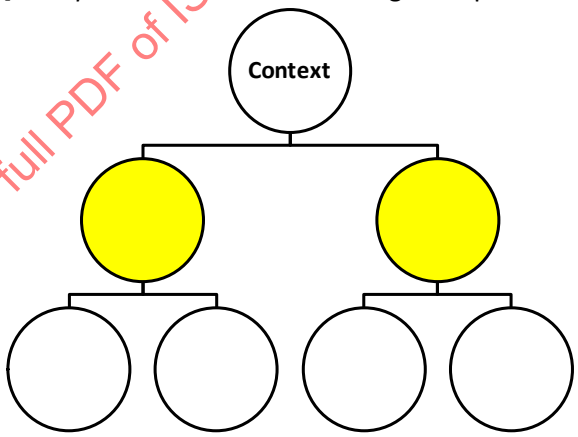
This simple type defines different node sets in relation to the current context node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

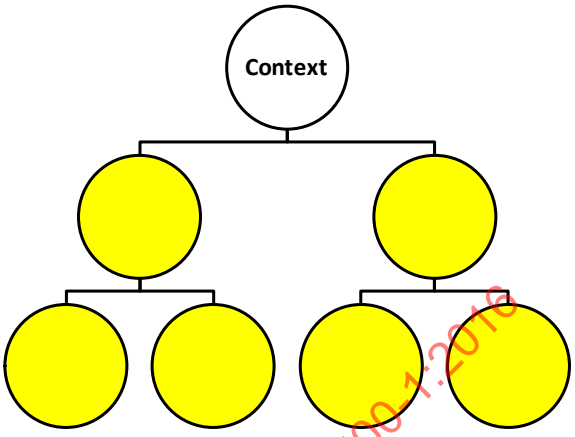
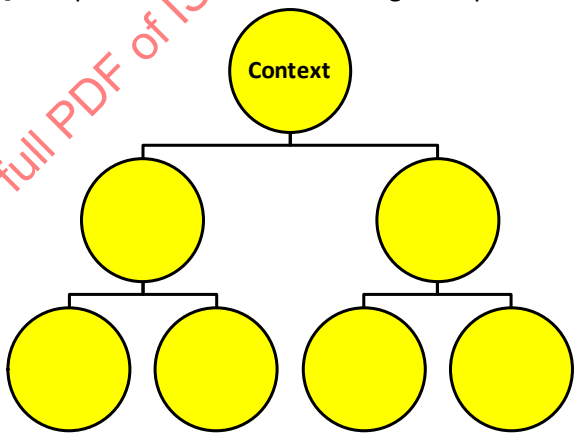
Enumeration Value	Description
ancst (Ancestor)	Specifies a set of nodes between the current context node and the root node, including the root node. [*Example*: Consider the following example  *end example*]
ancstOrSelf (Ancestor or Self)	Specifies a set of nodes between the current context node and the root node, including the root node and the context node.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	[*Example*: Consider the following example  *end example*]
ch (Child)	Specifies a set of children of the current context node. [*Example*: Consider the following example  *end example*]
des (Descendant)	Specifies a set of all nodes beneath the current context node. [*Example*: Consider the following example

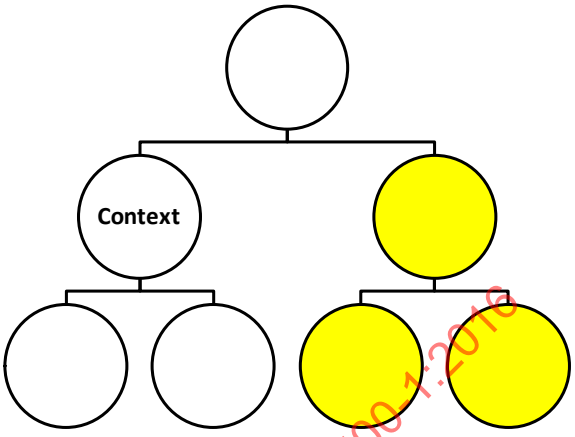
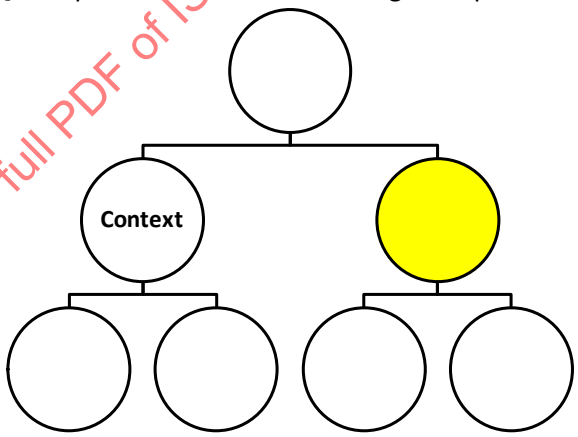
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
desOrSelf (Descendant or Self)	Specifies a set of all nodes beneath the current context node, including the context node. [*Example*: Consider the following example  *end example*]
follow (Follow)	Specifies the set of nodes which are peers after the context node and all descendants of the peers. [*Example*: Consider the following example

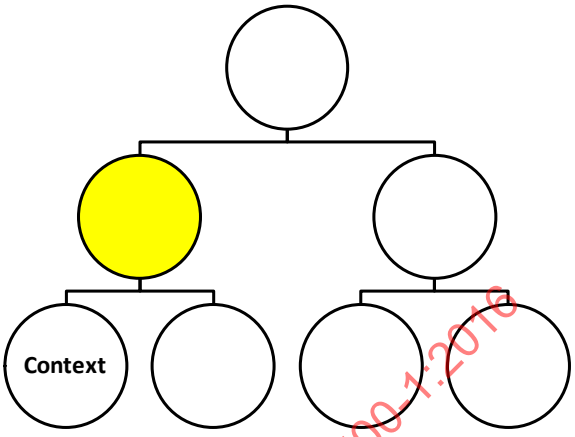
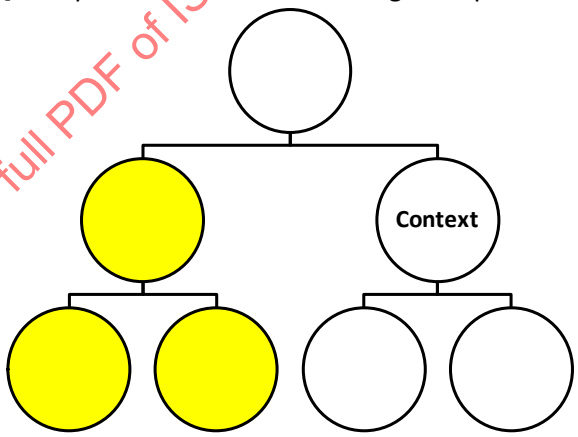
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
followSib (Follow Sibling)	Specifies the set of nodes which are peers after the context node. [*Example*: Consider the following example  *end example*]
none (None)	Specifies no node.
par (Parent)	Specifies the parent node. [*Example*: Consider the following example

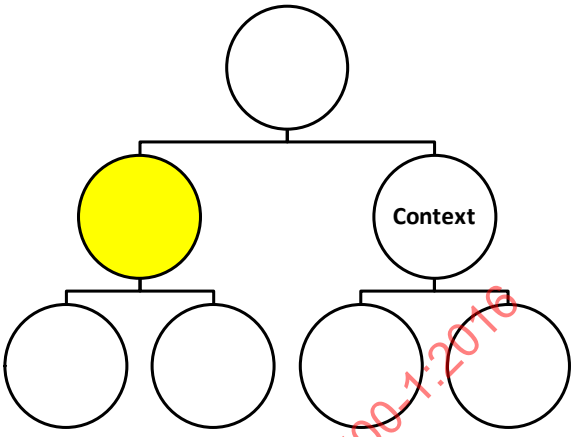
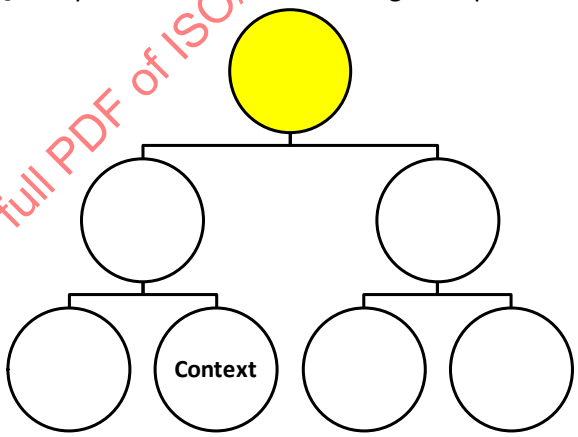
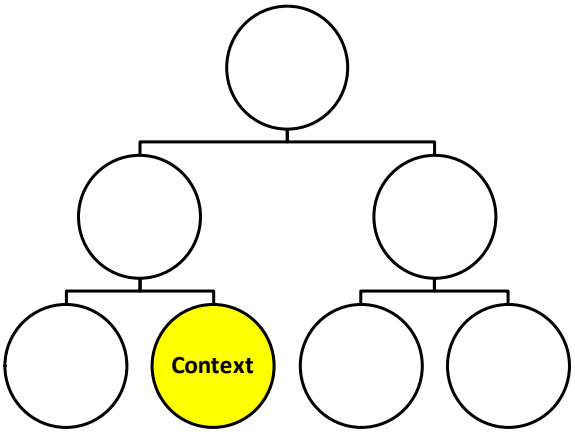
IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
preced (Preceding)	Specifies the set of nodes which are peers before the context node and all the descendants of the peers. [*Example*: Consider the following example  *end example*]
precedSib (Preceding Sibling)	Specifies the set of nodes which are peers before the context node. [*Example*: Consider the following example

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
root (Root)	Specifies the top-most node of the diagram. [*Example*: Consider the following example  *end example*]
self (Self)	Specifies the calling context node. [*Example*: Consider the following example  *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisType) is located in §A.5.3. *end note*]

21.4.7.7 ST_AxisTypes (Axis Type List)

This simple type represents a list of axis types.

This simple type allows a list of items of the ST_AxisType simple type (§21.4.7.6).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_AxisTypes) is located in §A.5.3. *end note*]

21.4.7.8 ST_BendPoint (Bend Point)

This simple type defines where a bend is to occur within a connection between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
beg (Beginning)	The bend is to occur at the beginning of the connection.
def (Default)	The default bend is used. By default connections bend in the center.
end (End)	The bend is to occur at the end of the connection.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BendPoint) is located in §A.5.3. *end note*]

21.4.7.9 ST_Booleans (Boolean List.)

A list of booleans.

This simple type allows a list of items of the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Booleans) is located in §A.5.3. *end note*]

21.4.7.10 ST_BoolOperator (Boolean Constraint)

This simple type specified Boolean operations which can be applied to compare constraints.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
equ (Equal)	Equal operator.
gte (Greater Than or Equal to)	Specifies the greater than or equal to Boolean operator.
lte (Less Than or Equal to)	Specifies the less than or equal to Boolean operator.
none (None)	Specifies a none Boolean operator

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BoolOperator) is located in §A.5.3. *end note*]

21.4.7.11 ST_Breakpoint (Breakpoint)

This simple type defines at what point the wrapping of nodes occurs for the snake algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bal (Balanced)	Specifies that the number of nodes in every row and every column should be equal.
endCnv (End of Canvas)	Specifies that nodes are added to the next column or row after filling the current column or row's space.
fixed (Fixed)	Specifies to use a user defined number of nodes in a column or row.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Breakpoint) is located in §A.5.3. *end note*]

21.4.7.12 ST_CenterShapeMapping (Center Shape Mapping)

This simple type defines the behavior of the cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fNode (First Node)	Specifies a node which is always in the center of a cycle diagram.
none (None)	Specifies the normal layout of a cycle diagram.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CenterShapeMapping) is located in §A.5.3. *end note*]

21.4.7.13 ST_ChildAlignment (Child Alignment)

This simple type defines how to align a node in its allocated space.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies to align the node to the bottom.
l (Left)	Specifies to align the node to the left.
r (Right)	Specifies to align the node to the right.
t (Top)	Specifies to align the node to the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildAlignment) is located in §A.5.3. *end note*]

21.4.7.14 ST_ChildDirection (Child Direction)

This simple type defines the layout direction of child nodes related to a specific parent node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
horz (Horizontal)	Specifies that the child nodes are to be laid out horizontally.
vert (Vertical)	Specifies that the child nodes are to be laid out vertically.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildDirection) is located in §A.5.3. *end note*]

21.4.7.15 ST_ChildOrderType (Child Order)

This simple type specifies the child order for a given layout node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Child order along the bottom.
t (Top)	Top child order.

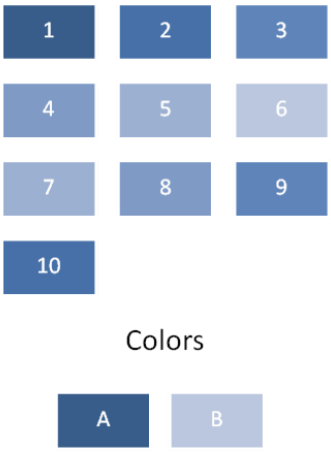
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ChildOrderType) is located in §A.5.3. *end note*]

21.4.7.16 ST_ClrAppMethod (Color Application Method Type)

This simple type defines the way a given set of colors is applied to a set of nodes or items across a diagram.

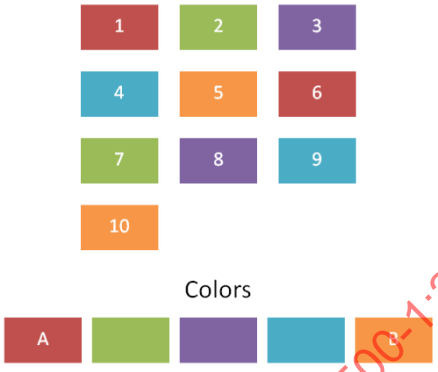
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cycle (Cycle)	The colors apply from A to B to A if A and B were the colors present. [*Example*: Consider the following image as an example of cycle applied to a diagram:  In this example, the color A is applied to node 1 and node 10. Color B is considered the node color between A and A across the diagram. Colors interpolate across the diagram from A to B back to A. *end example*]
repeat (Repeat)	The colors apply from A through B to A through B if A through B were the colors present. [*Example*: Consider the following image as an example

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	of repeat applied to a diagram:  In this example, the color A is applied to node 1, the next color to node 2, and so on through color B, then this coloring is repeated until there are no more nodes to color. *end example*]
span (Span)	The colors interpolate from A to B across the entire diagram if A and B were the colors present. [*Example*: Consider the following image as an example of span applied to a diagram: In this example, the color A is applied to node 1, the color B is applied to node 10 and the colors applied to nodes 2 through 9 are interpolated between colors A and B. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ClrAppMethod) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

21.4.7.17 ST_ConnectorDimension (Connector Dimension)

This simple type defines the dimensionality of the connection between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1D (1 Dimension)	Specifies a one dimensional connection, or rather a line.
2D (2 Dimensions)	Specifies a two dimensional connection which has both width and height.
cust (Custom)	Specifies a custom connection type.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorDimension) is located in §A.5.3. *end note*]

21.4.7.18 ST_ConnectorPoint (Connector Point)

This simple type defines different connection sites available on a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
auto (Auto)	Specifies that the algorithm determines the best connection site to use.
bCtr (Bottom Center)	Specifies that the bottom, center connection site is to be used.
bL (Bottom Left)	Specifies that the bottom, left connection site is to be used.
bR (Bottom Right)	Specifies that the bottom right connection site is to be used.
ctr (Center)	Specifies that the center connection site is to be used.
midL (Middle Left)	Specifies that the middle left connection site is to be used.
midR (Middle Right)	Specifies that the middle right connection site is to be used.
radial (Radial)	Specifies connections along a radial path to support the use of connections in cycle diagrams.
tCtr (Top Center)	Specifies that the top center connection site is to be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
tL (Top Left)	Specifies that the top left connection site is to be used.
tR (Top Right)	Specifies that the top right connection site is to be used.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorPoint) is located in §A.5.3. *end note*]

21.4.7.19 ST_ConnectorRouting (Connector Routing)

This simple type defines how the routing of a connection between two nodes is supposed to progress from node 1 to node 2.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bend (Bending)	Specifies a bending connection which bends at a right angle.
curve (Curve)	Specifies a connection which is curved.
longCurve (Long Curve)	Specifies a connection that is curved that has a greater radius than a simple curved connection.
stra (Straight)	Specifies a straight connection.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConnectorRouting) is located in §A.5.3. *end note*]

21.4.7.20 ST_ConstraintRelationship (Constraint Relationship)

This simple type specifies the types of constraint relationships which are present and can be used.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ch (Child)	The constraint should reference a child node.
des (Descendant)	The layout node can map to the descendants of the data point.
self (Self)	The layout node maps to the current data point.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConstraintRelationship) is located in §A.5.3. *end note*]

21.4.7.21 ST_ConstraintType (Constraint Type)

This simple type defines the list of possible constraints available for use.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alignOff (Alignment Offset)	This value defines the alignment offset for a node.
b (Bottom)	The bottom of the node.
begMarg (Beginning Margin)	Specifies the beginning margin.
begPad (Beginning Padding)	Specifies the beginning padding.
bendDist (Bending Distance)	Specifies the distance from the start of a connector to a bend in the connector.
bMarg (Bottom Margin)	Specifies the bottom margin.
bOff (Bottom Offset)	Specifies the bottom offset.
connDist (Connection Distance)	Specifies the connection distance.
ctrX (Center Height)	Specifies the center of the height.
ctrXOff (Center X Offset)	Specifies the center x coordinate offset.
ctrY (Center Width)	Specifies the center of the width.
ctrYOff (Center Y Offset)	Specifies the center y coordinate offset.
diam (Diameter)	Specifies the diameter.
endMarg (End Margin)	Specifies the ending margin.
endPad (End Padding)	Specifies the end padding.
h (Height)	Specifies the height.
hArH (Arrowhead Height)	Specifies the height of the arrowhead portion of the connector.
hOff (Height Offset)	Specifies the amount to offset the height.
l (Left)	Specifies the left constraint.
lMarg (Left Margin)	Specifies the left margin.
lOff (Left Offset)	Specifies the left offset.
none (Unknown)	Unknown constraint.
primFontSz (Primary Font Size)	The primary font size.
pyraAcctRatio (Pyramid Accent Ratio)	Specifies the fraction of the width of the diagram that is reserved for the fly outs at their shortest distance.
r (Right)	Specifies the right constraint.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
rMarg (Right Margin)	Specifies the right margin constraint.
rOff (Right Offset)	Specifies the right offset constraint.
secFontSz (Secondary Font Size)	The secondary font size.
secSibSp (Secondary Sibling Spacing)	The secondary sibling spacing.
sibSp (Sibling Spacing)	Specifies the minimum distance between sibling shapes.
sp (Spacing)	Specifies the spacing defined.
stemThick (Stem Thickness)	Specifies the thickness of the arrow's shaft.
t (Top)	Specifies the top constraint.
tMarg (Top Margin)	Top margin constraint.
tOff (Top Offset)	Top offset constraint.
userA (User Defined A)	User defined information.
userB (User Defined B)	User defined information.
userC (User Defined C)	User defined information.
userD (User Defined D)	User defined information.
userE (User Defined E)	User defined information.
userF (User Defined F)	User defined information.
userG (User Defined G)	User defined information.
userH (User Defined H)	User defined information.
userI (User Defined I)	User defined information.
userJ (User Defined J)	User defined information.
userK (User Defined K)	User defined information.
userL (User Defined L)	User defined information.
userM (User Defined M)	User defined information.
userN (User Defined N)	User defined information.
userO (User Defined O)	User defined information.
userP (User Defined P)	User defined information.
userQ (User Defined Q)	User defined information.
userR (User Defined R)	User defined information.
userS (User Defined S)	User defined information.
userT (User Defined T)	User defined information.
userU (User Defined U)	User defined information.
userV (User Defined V)	User defined information.
userW (User Defined W)	User defined information.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
userX (User Defined X)	User defined information.
userY (User Defined Y)	User defined information.
userZ (User Defined Z)	User defined information.
w (Width)	The width parameter.
wArH (Arrowhead Width)	Specifies the width of the arrowhead portion of the connector.
wOff (Width Offset)	Offsets the width by the specified amount.

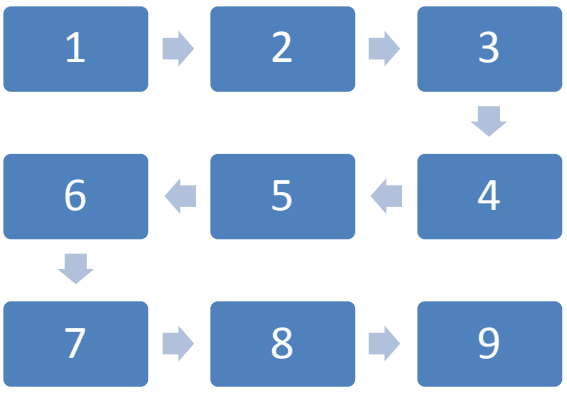
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConstraintType) is located in §A.5.3. *end note*]

21.4.7.22 ST_ContinueDirection (Continue Direction)

This simple type specifies the behavior of the direction that additional nodes are added to new rows or columns in the snake algorithm.

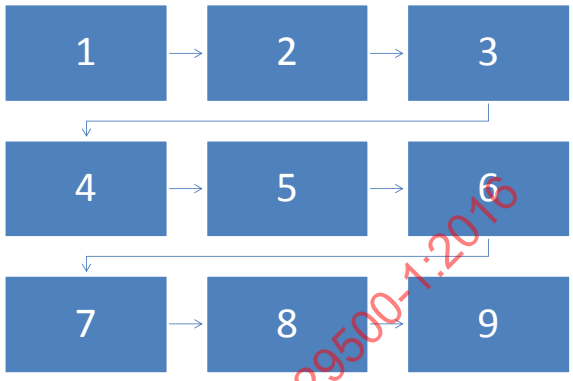
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
revDir (Reverse Direction)	Specifies that the direction is to be revered on a subsequent row or column. [*Example*: Consider the following diagram as an example of reverse direction  *end example*]
sameDir (Same Direction)	Specifies that the direction is to be maintained on a subsequent row or column.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	[*Example*: Consider the following diagram as an example of same direction  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ContinueDirection) is located in §A.5.3. *end note*]

21.4.7.23 ST_CxnType (Connection Type)

This simple type defines the different types of relationships that can be defined between two nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
parOf (Parent Of)	This defines a parent-child relationship in the sense that node X is a parent of node Y.
presOf (Presentation Of)	A presentation type relationship. This type of relationship exists to actually present data.
presParOf (Presentation Parent Of)	A relationship defining a parent of a presentation node.
unknownRelationship (Unknown Relationship)	The type of relationship is unknown.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CxnType) is located in §A.5.3. *end note*]

21.4.7.24 ST_DiagramHorizontalAlignment (Horizontal Alignment)

This simple type defines the horizontal alignment.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center alignment.
l (Left)	Specifies left alignment.
none (None)	Specifies no alignment defined.
r (Right)	Specifies right alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DiagramHorizontalAlignment) is located in §A.5.3. *end note*]

21.4.7.25 ST_DiagramTextAlignment (Text Alignment)

This simple type defines alignment types for text within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center aligned text.
l (Left)	Specifies left aligned text.
r (Right)	Specifies right aligned text.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_DiagramTextAlignment) is located in §A.5.3. *end note*]

21.4.7.26 ST_Direction (Diagram Direction Definition)

This simple type defines the possible values for a diagram's direction when displayed in an application.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
norm (Normal Direction)	This value specifies that the direction of the diagram should not be switched.
rev (Reversed Direction)	This value specifies that the direction of the diagram should be switched.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Direction) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

21.4.7.27 ST_ElementType (Data Point Type)

This simple type defines the different types of data points which are supported.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
all (All)	Defined as utilizing all of the nodes.
asst (Assistant)	The assistant nodes.
doc (Document)	Specifies the a node on the document level.
node (Node)	Data nodes that are children of other data nodes.
nonAsst (Non Assistant)	Selects all of the non-assistant nodes.
nonNorm (Non Normal)	Selects the non-normal elements.
norm (Normal)	Selects a normal elements.
parTrans (Parent Transition)	The transition associated with the parent node.
pres (Presentation)	This refers to a presentation node.
sibTrans (Sibling Transition)	Use only sibling transitions between data nodes. These transitions represent sibling relationships between nodes, and are frequently mapped to arrows between shapes in the drawing. A sibTrans value is sometimes used to create white space between nodes.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ElementType) is located in §A.5.3. *end note*]

21.4.7.28 ST_ElementTypes (Diagream Layout Node Type List)

A list of diagram layout node types.

This simple type allows a list of items of the ST_ElementType simple type (§21.4.7.27).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ElementTypes) is located in §A.5.3. *end note*]

21.4.7.29 ST_FallbackDimension (Fallback Dimension)

Specifies the dimensionality by which nodes can grow or shrink automatically.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
1D (1 Dimension)	Specifies that the node can grow or shrink by its height

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	or its width, but not both.
2D (2 Dimensions)	Specifies that the node can grow or shrink by both height and width.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FallbackDimension) is located in §A.5.3. *end note*]

21.4.7.30 ST_FlowDirection (Flow Direction)

This simple type defines how the progression of new nodes are to be entered into the diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
col (Column)	Specifies that the layout occurs in a column-based fashion. This would mean laying out the nodes from top to bottom, before moving left to right.
row (Row)	Specifies that the layout occurs in a row-based fashion. This would mean laying out the nodes from left to right before moving from top to bottom.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FlowDirection) is located in §A.5.3. *end note*]

21.4.7.31 ST_FunctionArgument (Function Argument)

Conditional expression function argument.

This simple type is a union of the following types:

- The ST_VariableType simple type (§21.4.7.64).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionArgument) is located in §A.5.3. *end note*]

21.4.7.32 ST_FunctionOperator (Function Operator)

This simple type defines the condition expression functions which can be used to perform operations.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
equ (Equal)	Equal function operator.
gt (Greater Than)	Specifies the grater than function operator.
gte (Greater Than or Equal to)	Specifies the greater than or equal to function operator.
lt (Less Than)	Specifies the less than function operator.
lte (Less Than or Equal to)	Specifies the less than or equal to function operator.
neq (Not Equal To)	Specifies the not equal to function operator.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionOperator) is located in §A.5.3. *end note*]

21.4.7.33 ST_FunctionType (Function Type)

This simple type defines the set of available conditional expression function types present for use.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
cnt (Count)	Specifies a count.
depth (Depth)	Specifies the depth.
maxDepth (Max Depth)	Defines the maximum depth.
pos (Position)	Retrieves the position of the node in the specified set of nodes.
posEven (Position Even)	Returns 1 if the specified node is at an even numbered position in the data model.
posOdd (Position Odd)	Returns 1 if the specified node is in an odd position in the data model.
revPos (Reverse Position)	Reverse position function.
var (Variable)	Used to reference a variable.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionType) is located in §A.5.3. *end note*]

21.4.7.34 ST_FunctionValue (Function Value)

Conditional expression function value.

This simple type is a union of the following types:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- The ST_AnimLvlStr simple type (§21.4.7.2).
- The ST_AnimOneStr simple type (§21.4.7.3).
- The ST_Direction simple type (§21.4.7.26).
- The ST_HierBranchStyle simple type (§21.4.7.37).
- The ST_ResizeHandlesStr simple type (§21.4.7.54).
- The W3C XML Schema boolean datatype.
- The W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FunctionValue) is located in §A.5.3. *end note*]

21.4.7.35 ST_GrowDirection (Grow Direction)

This simple type defines different starting locations for nodes within the snake algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bL (Bottom Left)	Specifies the placement of nodes is to start in the bottom left corner.
bR (Bottom Right)	Specifies the placement of nodes is to start in the bottom right corner.
tL (Top Left)	Specifies the placement of nodes is to start in the top left corner.
tR (Top Right)	Specifies the placement of nodes is to stat in the top right corner.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_GrowDirection) is located in §A.5.3. *end note*]

21.4.7.36 ST_HierarchyAlignment (Hierarchy Alignment)

This simple type defines different relative locations of child nodes and their descendants to a parent node within a hierarchy diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bCtrCh (Bottom Center Child)	Specifies the child nodes are placed below the parent node and that they are center aligned to the parent node.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
bCtrDes (Bottom Center Descendant)	Specifies the descendant nodes are placed below the parent node and that they are center aligned to the parent node.
bL (Bottom Left)	Specifies the child and descendant nodes are placed below the parent node and that the set is left aligned.
bR (Bottom Right)	Specifies the child and descendant nodes are placed below the parent node and the set is right aligned.
lB (Left Bottom)	Specifies the child and descendant nodes are placed to the left of the parent node and that the set is bottom aligned.
lCtrCh (Left Center Child)	Specifies the child nodes are placed to the left of the parent node and that the set is center aligned.
lCtrDes (Left Center Descendant)	Specifies the descendant nodes are placed to the left of the parent node and that the set is center aligned.
lT (Left Top)	Specifies the child and descendant nodes are placed to the left of the parent node and that the set is top aligned.
rB (Right Bottom)	Specifies the child and descendant nodes are placed to the right of the parent node and that the set is bottom aligned.
rCtrCh (Right Center Children)	Specifies the child nodes are placed to the right of the parent node and that the set is center aligned.
rCtrDes (Right Center Descendants)	Specifies the descendant nodes are placed to the right of the parent node and that the set is center aligned.
rT (Right Top)	Specifies the child and descendant nodes are placed to the right of the parent node and that the set is top aligned.
tCtrCh (Top Center Children)	Specifies the child nodes are placed above the parent node and that the set is center aligned.
tCtrDes (Top Center Descendants)	Specifies the descendant nodes are placed above the parent node and that the set is center aligned.
tL (Top Left)	Specifies the child and descendant nodes are placed above the parent node and that the set is left aligned.
tR (Top Right)	Specifies the child and descendant nodes are placed above the parent node and that the set is right aligned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

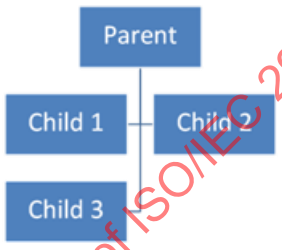
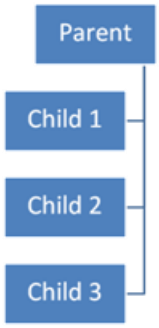
[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HierarchyAlignment) is located in §A.5.3. *end note*]

21.4.7.37 ST_HierBranchStyle (Hierarchy Branch Style Definition)

This simple type specifies the possible values for the branch style of a hierarchy diagram.

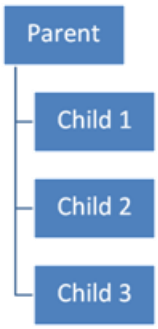
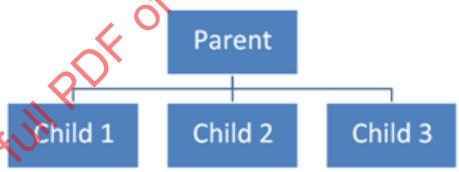
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
hang (Hanging)	The branch style is hanging from the parent. [*Example*: Consider the following image as an example of a hanging branch style:  *end example*]
init (Initial)	This means that the value has not been set.
l (Left)	The branch style falls off the left. [*Example*: Consider the following image as an example of a left hanging branch style:  *end example*]
r (Right)	The branch style falls off the right. [*Example*: Consider the following image as an example of a right hanging branch style:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
std (Standard)	The standard branch style is to be used. [*Example*: Consider the following image as an example of a standard hanging branch style:  *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HierBranchStyle) is located in §A.5.3. *end note*]

21.4.7.38 ST_HueDir (Hue Direction)

When given two colors to interpolate between, one can go in one of two directions around the color wheel to perform the interpolation. This simple type defines that direction.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ccw (Counterclockwise Hue Direction)	A hue interpolation in the counterclockwise direction.
cw (Clockwise Hue Direction)	A hue interpolation in the clockwise direction.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HueDir) is located in §A.5.3. *end note*]

21.4.7.39 ST_Index1 (1-Based Index)

A 1-based index.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Index1) is located in §A.5.3. *end note*]

21.4.7.40 ST_Ints (Integer List)

A list of integers.

This simple type allows a list of items of the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Ints) is located in §A.5.3. *end note*]

21.4.7.41 ST_LayoutShapeType (Layout Shape Type)

All of the available shape types.

This simple type is a union of the following types:

- The ST_OutputShapeType simple type (§21.4.7.48).
- The ST_ShapeType simple type (§20.1.10.56).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LayoutShapeType) is located in §A.5.3. *end note*]

21.4.7.42 ST_LinearDirection (Linear Direction)

This simple type defines the direction of growth of new nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies growth to start from the bottom.
fromL (From Left)	Specifies growth to start from the left.
fromR (From Right)	Specifies growth to start from the right.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
fromT (From Top)	Specifies growth to start from the Top

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LinearDirection) is located in §A.5.3. *end note*]

21.4.7.43 ST_ModelId (Model Identifier)

The unique ID of the element within the data model. Model Identifiers can be either longs or guids.

This simple type is a union of the following types:

- The ST_Guid simple type (§22.9.2.4).
- The W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ModelId) is located in §A.5.3. *end note*]

21.4.7.44 ST_NodeCount (Number of Nodes Definition)

This simple type defines a count of the number of nodes for a property in a diagram. A value of -1 shall mean that the value is unbounded.

This simple type's contents are a restriction of the W3C XML Schema int datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -1.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeCount) is located in §A.5.3. *end note*]

21.4.7.45 ST_NodeHorizontalAlignment (Node Horizontal Alignment)

This simple type defines the horizontal alignment of a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies center alignment.
l (Left)	Specifies left alignment.
r (Right)	Specifies right alignment.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeHorizontalAlignment) is located in §A.5.3. *end note*]

21.4.7.46 ST_NodeVerticalAlignment (Node Vertical Alignment)

This simple type defines the vertical alignment of a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies bottom alignment.
mid (Middle)	Specifies middle alignment.
t (Top)	Specifies top alignment.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_NodeVerticalAlignment) is located in §A.5.3. *end note*]

21.4.7.47 ST_Offset (Offset)

This simple type defines whether or not subsequent rows or columns in the snake algorithm are offset from the preceding row or column.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies no offset.
off (Offset)	Specifies that the nodes are shifted by some amount relative to the preceding row or column.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Offset) is located in §A.5.3. *end note*]

21.4.7.48 ST_OutputShapeType (Output Shape Type)

Shapes which are special specifically for a DrawingML diagram.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
conn (Connection)	Connection shape type.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
none (None)	None.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OutputShapeType) is located in §A.5.3. *end note*]

21.4.7.49 ST_ParameterId (Parameter Identifier)

This simple type defines algorithm parameters which can be modified in order to adjust the behavior of algorithms for use in layout nodes.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alignTx (Text Alignment)	This value defines how the text is aligned in a node.
ar (Aspect Ratio)	Specifies the aspect ratio (width to height) of the composite node to use when determining child constraints. A value of 0 means leave the width and height constraints as is. The algorithm can temporarily shrink one dimension to achieve that ratio. [*Example*: If a composite node has a width constraint of 20 and height constraint of 10, and if ar=1.5, composite uses a width value of 15 to calculate the composite node's child constraints. However, the algorithm does not propagate this value to other nodes. *end example*]
autoTxRot (Auto Text Rotation)	Auto text rotation.
begPts (Beginning Points)	Beginning Points
begSty (Beginning Arrowhead Style)	Beginning Arrowhead Style
bendPt (Bend Point)	The bend point.
bkpt (Breakpoint)	Specifies the point at which the diagram starts to snake. The value bal specifies that snaking begin at an even number of rows and columns. The value fixed specifies that snaking begin at a fixed point. The value endCnv specifies that snaking begin when there is no more room for a shape in the row.
bkPtFixedVal (Breakpoint Fixed Value)	Specifies where the snake should break, if bkpt=fixed.
chAlign (Child Alignment)	Specifies the alignment of the children.
chDir (Child Direction)	The child direction.
connRout (Connection Route)	The route of the connection.
contDir (Continue Direction)	Specifies the direction of the subsequent row or column. [*Example*: If the algorithm initially places the

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	nodes from left to right, revDir places the nodes in the next row from right to left. However if the algorithm uses contDir, the nodes on the next row are arranged from left to right. *end example*]
ctrShpMap (Center Shape Mapping)	Specifies where to place nodes in relation to the center circle.
dim (Connector Dimension)	Specifies the connector dimension.
dstNode (Destination Node)	Specifies the name of the layout node from which to end the connection from.
endPts (End Points)	Specifies the end points.
endSty (End Style)	Specifies the end style.
fallback (Fallback Scale)	1D specifies fallback. It only scales in one dimension. 2D specifies fallback. It scales in both dimensions equally.
flowDir (Flow Direction)	Specifies whether nodes are arranged in rows or columns.
grDir (Grow Direction)	Specifies from which corner the snake grows. [*Example*: If the algorithm uses a top left value, the snake grows from the top left. *end example*]
hierAlign (Hierarchy Alignment)	The alignment of the hierarchy.
horzAlign (Horizontal Alignment)	Aligns all the child nodes within the space reserved for the parent and adjusts child positions in the x direction.
linDir (Linear Direction)	Specifies the linear direction.
lnSpAfChP (Line Spacing After Children Paragraph)	Line spacing after children.
lnSpAfParP (Line Spacing After Parent Paragraph)	Line spacing after the parent.
lnSpCh (Line Spacing Children)	Line spacing of the children
lnSpPar (Line Spacing Parent)	Line spacing of the parent.
nodeHorzAlign (Node Horizontal Alignment)	Specifies how child nodes are aligned within the extents of the canvas. [*Example*: You can align the tops of all the child nodes, but center all of them within the canvas. *end example*]
nodeVertAlign (Node Vertical Alignment)	Specifies how child nodes are aligned within the extents of the canvas. Same as nodeHorzAlign, but in the y direction.
off (Offset)	Specifies the offset.
parTxLTRAlign (Parent Text Left-to-Right Alignment)	Specifies the paragraph alignment of parent text when the shape has only parent text. This parameter applies when the text direction is left to right.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
parTxRTLAlign (Parent Text Right-to-Left Alignment)	Specifies the paragraph alignment of parent text when the shape has only parent text. This parameter applies when the text direction is right to left.
pyraAcctBkgdNode (Pyramid Accent Background Node)	If pyramid has a composite child node, specifies the name of the node that is a child of the composite that makes up the child flyout shape. If the node specifies a shape of the nonIsoscelesTrapezoid autoshape, it modifies the adjust handles in order to fit the flyout flush against the side of the pyramid.
pyraAcctPos (Pyramid Accent Position)	Specifies the placement of the flyout grandchildren.
pyraAcctTxMar (Pyramid Accent Text Margin)	Specifies the placement of one edge of the child text (grandchild node). If the value is step, the text is against the edge of the pyramid. If the value is stack, the text aligns.
pyraAcctTxNode (Pyramid Accent Text Node)	If pyramid has a composite child node, specifies the child node that should hold the child text.
pyraLvlNode (Pyramid Level Node)	If pyramid has a composite child node, specifies the name of the node that is a child of the composite that makes up the pyramid itself. If the node specifies a trapezoid shape, it modifies the adjustment handles to construct a pyramid.
rotPath (Rotation Path)	The rotation path specified.
rtShortDist (Route Shortest Distance)	If true, the connector is routed through the shortest distance between the points.
secChAlign (Secondary Child Alignment)	The secondary child alignment.
secLinDir (Secondary Linear Direction)	The secondary linear direction.
shpTxLTRAlignCh (Shape Text Left-to-Right Alignment)	Specifies the paragraph alignment of all text within the shape when the shape contains both parent and child text. This parameter applies when the text direction is left to right.
shpTxRTLAlignCh (Shape Text Right-to-Left Alignment)	Specifies the paragraph alignment of all text within the shape when the shape contains both parent and child text. This parameter applies when the text direction is right to left.
spanAng (Span Angle)	Specifies the angle the cycle spans. Final shapealign text is placed at stAng+spanAng, unless spanAng=360. In that case, the algorithm places the text so that shapes do not overlap.
srcNode (Source Node)	Specifies the name of the layout node from which to start the connection.
stAng (Start Angle)	Specifies the angle at which the first shape is placed.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	Angles are in degrees, measured clockwise from a line pointing straight upward from the center of the cycle.
stBulletLvl (Start Bullets At Level)	Specifies whether bullets start at the top level (1) or with children (2).
stElem (Start Element)	Specifies the point type of the layout node to use as the first shape in the cycle.
txAnchorHorz (Text Anchor Horizontal)	Specifies the y-axis position of the text area within a shape.
txAnchorHorzCh (Text Anchor Horizontal With Children)	Specifies that the definition can allow a different text anchoring on the x-axis, if child nodes exist in the shape.
txAnchorVert (Text Anchor Vertical)	Specifies the x-axis position of the text area within a shape.
txAnchorVertCh (Text Anchor Vertical With Children)	Specifies that the definition can allow a different text anchoring on the y-axis, if child nodes exist in the shape.
txBlDir (Text Block Direction)	Specifies whether the text block is vertical or horizontal.
txDir (Text Direction)	Specifies where the text of the first node starts.
vertAlign (Vertical Alignment)	Aligns all the child nodes within the space reserved for the parent and adjusts child positions in the y direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ParameterId) is located in §A.5.3. *end note*]

21.4.7.50 ST_ParameterVal (Parameter Values)

Specifies the list of parameter types that can be used by a diagram.

This simple type is a union of the following types:

- The ST_ArrowheadStyle simple type (§21.4.7.4).
- The ST_AutoTextRotation simple type (§21.4.7.5).
- The ST_BendPoint simple type (§21.4.7.8).
- The ST_Breakpoint simple type (§21.4.7.11).
- The ST_CenterShapeMapping simple type (§21.4.7.12).
- The ST_ChildAlignment simple type (§21.4.7.13).
- The ST_ChildDirection simple type (§21.4.7.14).
- The ST_ConnectorDimension simple type (§21.4.7.17).
- The ST_ConnectorPoint simple type (§21.4.7.18).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- The ST_ConnectorRouting simple type (§21.4.7.19).
- The ST_ContinueDirection simple type (§21.4.7.22).
- The ST_DiagramHorizontalAlignment simple type (§21.4.7.24).
- The ST_DiagramTextAlignment simple type (§21.4.7.25).
- The ST_FallbackDimension simple type (§21.4.7.29).
- The ST_FlowDirection simple type (§21.4.7.30).
- The ST_GrowDirection simple type (§21.4.7.35).
- The ST_HierarchyAlignment simple type (§21.4.7.36).
- The ST_LinearDirection simple type (§21.4.7.42).
- The ST_NodeHorizontalAlignment simple type (§21.4.7.45).
- The ST_NodeVerticalAlignment simple type (§21.4.7.46).
- The ST_Offset simple type (§21.4.7.47).
- The ST_PyramidAccentPosition simple type (§21.4.7.52).
- The ST_PyramidAccentTextMargin simple type (§21.4.7.53).
- The ST_RotationPath simple type (§21.4.7.55).
- The ST_SecondaryChildAlignment simple type (§21.4.7.56).
- The ST_SecondaryLinearDirection simple type (§21.4.7.57).
- The ST_StartingElement simple type (§21.4.7.58).
- The ST_TextAnchorHorizontal simple type (§21.4.7.59).
- The ST_TextAnchorVertical simple type (§21.4.7.60).
- The ST_TextBlockDirection simple type (§21.4.7.61).
- The ST_TextDirection simple type (§21.4.7.62).
- The ST_VerticalAlignment simple type (§21.4.7.65).
- The W3C XML Schema boolean datatype.
- The W3C XML Schema double datatype.
- The W3C XML Schema int datatype.
- The W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ParameterVal) is located in §A.5.3. *end note*]

21.4.7.51 ST_PtType (Point Type)

This simple type defines the different point types which can be utilized to create diagrams in DiagramML.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
asst (Assistant Element)	This point type is used in a hierarchy diagram to represent an assistant element.
doc (Document)	This point type specifies a document type point. This

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	point type can be thought of as the root node associated with the document itself.
node (Node)	The node point type specifies a basic point type.
parTrans (Parent Transition)	This point type specifies a parent transition element.
pres (Presentation)	Specifies a presentation point type.
sibTrans (Sibling Transition)	This point type specifies a sibling transition element.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PtType) is located in §A.5.3. *end note*]

21.4.7.52 ST_PyramidAccentPosition (Pyramid Accent Position)

This simple type defines different positioning for the accent shapes which can be associated with a pyramid algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
aft (Pyramid Accent After)	Specifies that the accent shapes are to be placed to the right of the pyramid.
bef (Before)	Specifies that the accent shapes are to be placed to the left of the pyramid.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PyramidAccentPosition) is located in §A.5.3. *end note*]

21.4.7.53 ST_PyramidAccentTextMargin (Pyramid Accent Text Margin)

This simple type defines different ways to lay out text in the accent shape for a pyramid algorithm.

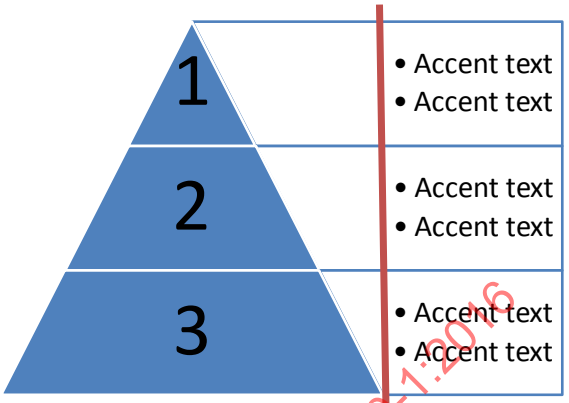
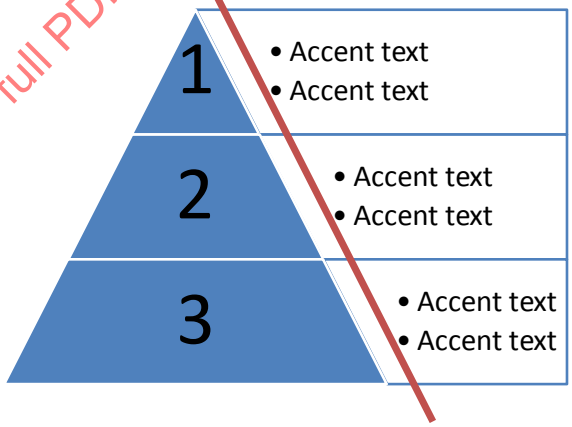
This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
stack (Stack)	Specifies that all accent shape text is to be left aligned. [*Example*: Consider the following example of a pyramid diagram

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	 *end example*]
step (Step)	Specifies that all accent shape text is to be relative to the pyramid. [*Example*: Consider the following example of a pyramid diagram  *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PyramidAccentTextMargin) is located in §A.5.3. *end note*]

21.4.7.54 ST_ResizeHandlesStr (Resize Handle)

This simple type defines the possible behaviors when resizing shapes within a diagram. Because the size of the shape plays a large role in the overall layout of other nodes within the diagram, there are two ways resize can occur on a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
exact (Exact)	This value specifies that the resize of the shape occurs and sizes exactly to the size the user defines, which causes all other shapes in the diagram to shrink or grow accordingly.
rel (Relative)	This value specifies that resize operations happen relatively. This means that the relative size difference between nodes is maintained before and after the resize operation.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ResizeHandlesStr) is located in §A.5.3. *end note*]

21.4.7.55 ST_RotationPath (Rotation Path)

This simple type defines rotation properties for nodes within the cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
alongPath (Along Path)	Specifies that the nodes should rotate in relation to their placement along the cycle.
none (None)	Specifies that the nodes should not rotate.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RotationPath) is located in §A.5.3. *end note*]

21.4.7.56 ST_SecondaryChildAlignment (Secondary Child Alignment)

This simple type defines different alignment properties of the both hanging layout type of the hierarchy algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies that the children nodes should be bottom aligned.
l (Left)	Specifies that the children nodes should be left aligned.
none (None)	Specifies no alignment.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
r (Right)	Specifies that the children nodes should be right aligned.
t (Top)	Specifies that the children nodes should be top aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondaryChildAlignment) is located in §A.5.3. *end note*]

21.4.7.57 ST_SecondaryLinearDirection (Secondary Linear Direction)

This simple type defines different directions for the nodes in a both hanging layout in the hierarchy algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies that the nodes begin from the bottom and move upward.
fromL (From Left)	Specifies that the nodes begin from the left and move right.
fromR (From Right)	Specifies that the nodes begin from the right and move left.
fromT (From Top)	Specifies that the nodes begin from the top and move downward.
none (None)	Specifies no direction.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SecondaryLinearDirection) is located in §A.5.3. *end note*]

21.4.7.58 ST_StartingElement (Starting Element)

This simple type defines behavior for the first node in a cycle algorithm.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
node (Node)	Specifies that a node should be placed first.
trans (Transition)	Specifies that a transition should be placed first.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_StartingElement) is located in §A.5.3. *end note*]

21.4.7.59 ST_TextAnchorHorizontal (Text Anchor Horizontal)

This simple type defines horizontal anchor points for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
ctr (Center)	Specifies text to be anchored to the center.
none (None)	Specifies no horizontal text anchor.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchorHorizontal) is located in §A.5.3. *end note*]

21.4.7.60 ST_TextAnchorVertical (Text Anchor Vertical)

This simple type defines vertical anchor points for text.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies text to be anchored to the bottom.
mid (Middle)	Specifies text to be anchored to the middle.
t (Top)	Specifies text to be anchored to the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextAnchorVertical) is located in §A.5.3. *end note*]

21.4.7.61 ST_TextBlockDirection (Text Block Direction)

This simple type defines different layout options for text within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
horz (Horizontal)	Specifies that the text is to be horizontal.
vert (Vertical Direction)	Specifies that the text is to be vertical.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextBlockDirection) is located in §A.5.3. *end note*]

21.4.7.62 ST_TextDirection (Text Direction)

This simple type defines different way the growth of additional text can occur within a node.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
fromB (From Bottom)	Specifies additional text grows from the bottom.
fromT (From Top)	Specifies additional text grows from the top.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TextDirection) is located in §A.5.3. *end note*]

21.4.7.63 ST_UnsignedInts (Unsigned Integer List)

A list of unsigned integers.

This simple type allows a list of items of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedInts) is located in §A.5.3. *end note*]

21.4.7.64 ST_VariableType (VariableType)

Conditional expression variable type.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
animLvl (Animation Level)	Specifies the animation level
animOne (Animate One)	Specifies animate as one.
bulEnabled (Bullets Enabled)	Specifies bullets enabled.
chMax (Child Max)	The maximum number of children.
chPref (Child Preference)	The preferred number of children.
dir (Direction)	Specifies the direction of the diagram.
hierBranch (Hierarchy Branch)	The hierarchy branch.
none (Unknown)	Unknown variable type.
orgChart (Organizational Chart Algorithm)	Algorithm that lays out an org chart.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
resizeHandles (Resize Handles)	Specifies the resize handles.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VariableType) is located in §A.5.3. *end note*]

21.4.7.65 ST_VerticalAlignment (Vertical Alignment)

This simple type defines different vertical alignment parameters.

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bottom)	Specifies bottom aligned.
mid (Middle)	Specifies middle aligned.
none (None)	Specifies no vertical alignment.
t (Top)	Specifies top aligned.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAlignment) is located in §A.5.3. *end note*]

21.4.7.66 ST_PrSetCustVal (Property Set Customized Value)

This simple type defines customization percentage values for certain elements in DrawingML.

This simple type is a union of the following types:

The ST_Percentage simple type (§22.9.2.9).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PrSetCustVal) is located in §A.5.3. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22. Shared MLs Reference Material

[No documentation has been entered for this section heading.]

22.1 Math

The following documentation specifies the XML representation of mathematical text for OOXML. This shared ML is known as that Office Math Markup Language (OMML). Mathematical text represented by OMML includes but is not limited to: equations, expressions, formulas, matrices, and other mathematical elements. The outermost OMML element of an instance of mathematical text in display mode is oMathPara, a math paragraph of one or more instances of mathematical text. Each instance of mathematical text inside the math paragraph is represented as a single oMath. Inside each oMath is a combination of mathematical runs (r) and objects or functions such as accents (acc) or fractions (f).

22.1.1 Table of Contents

This subclause is informative.

22.1.2 Elements 3602
22.1.2.1 acc (Accent) 3602
22.1.2.2 accPr (Accent Properties) 3603
22.1.2.3 aln (Alignment) 3603
22.1.2.4 alnScr (Align Scripts) 3605
22.1.2.5 argPr (Argument Properties) 3606
22.1.2.6 argSz (Argument Size) 3607
22.1.2.7 bar (Bar) 3609
22.1.2.8 barPr (Bar Properties) 3609
22.1.2.9 baseJc (Matrix Base Justification) 3610
22.1.2.10 begChr (Delimiter Beginning Character) 3613
22.1.2.11 borderBox (Border-Box Object) 3614
22.1.2.12 borderBoxPr (Border-Box Properties) 3614
22.1.2.13 box (Box Object) 3615
22.1.2.14 boxPr (Box Properties) 3615
22.1.2.15 brk (Break) 3616
22.1.2.16 brkBin (Break on Binary Operators) 3618
22.1.2.17 brkBinSub (Break on Binary Subtraction) 3619
22.1.2.18 cGp (Matrix Column Gap) 3620
22.1.2.19 cGpRule (Matrix Column Gap Rule) 3621
22.1.2.20 chr (Character) 3623
22.1.2.21 count (Matrix Column Count) 3624
22.1.2.22 cSp (Minimum Matrix Column Width) 3625
22.1.2.23 ctrlPr (Control Properties) 3626
22.1.2.24 d (Delimiter Object) 3627
22.1.2.25 defJc (Default Justification) 3628

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.1.2.26 deg (Degree) 3629
22.1.2.27 degHide (Hide Degree) 3630
22.1.2.28 den (Denominator) 3631
22.1.2.29 diff (Differential) 3631
22.1.2.30 dispDef (Use Display Math Defaults) 3633
22.1.2.31 dPr (Delimiter Properties) 3634
22.1.2.32 e (Element (Argument)) 3634
22.1.2.33 endChr (Delimiter Ending Character) 3636
22.1.2.34 eqArr (Array Object) 3636
22.1.2.35 eqArrPr (Array Properties) 3638
22.1.2.36 f (Fraction Object) 3639
22.1.2.37 fName (Function Name) 3640
22.1.2.38 fPr (Fraction Properties) 3640
22.1.2.39 func (Function Apply Object) 3641
22.1.2.40 funcPr (Function Properties) 3641
22.1.2.41 groupChr (Group-Character Object) 3642
22.1.2.42 groupChrPr (Group-Character Properties) 3643
22.1.2.43 grow (n-ary Grow) 3643
22.1.2.44 hideBot (Hide Bottom Edge) 3644
22.1.2.45 hideLeft (Hide Left Edge) 3645
22.1.2.46 hideRight (Hide Right Edge) 3646
22.1.2.47 hideTop (Hide Top Edge) 3647
22.1.2.48 interSp (Inter-Equation Spacing) 3647
22.1.2.49 intLim (Integral Limit Locations) 3648
22.1.2.50 intraSp (Intra-Equation Spacing) 3649
22.1.2.51 jc (Justification) 3649
22.1.2.52 lim (Limit) 3650
22.1.2.53 limLoc (n-ary Limit Location) 3650
22.1.2.54 limLow (Lower-Limit Object) 3651
22.1.2.55 limLowPr (Lower-Limit Properties) 3652
22.1.2.56 limUpp (Upper-Limit Object) 3652
22.1.2.57 limUppPr (Upper-Limit Properties) 3653
22.1.2.58 lit (Literal) 3654
22.1.2.59 lMargin (Left Margin) 3655
22.1.2.60 m (Matrix Object) 3656
22.1.2.61 mathFont (Math Font) 3658
22.1.2.62 mathPr (Math Properties) 3659
22.1.2.63 maxDist (Maximum Distribution) 3660
22.1.2.64 mc (Matrix Column) 3662
22.1.2.65 mcJc (Matrix Column Justification) 3663
22.1.2.66 mcPr (Matrix Column Properties) 3665
22.1.2.67 mcs (Matrix Columns) 3667
22.1.2.68 mPr (Matrix Properties) 3668
22.1.2.69 mr (Matrix Row) 3669
22.1.2.70 nary (n-ary Operator Object) 3671
22.1.2.71 naryLim (n-ary Limit Location) 3672
22.1.2.72 naryPr (n-ary Properties) 3673
22.1.2.73 noBreak (No Break) 3674

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.74 nor (Normal Text) 3675
22.1.2.75 num (Numerator) 3676
22.1.2.76 objDist (Object Distribution) 3677
22.1.2.77 oMath (Office Math) 3678
22.1.2.78 oMathPara (Office Math Paragraph) 3679
22.1.2.79 oMathParaPr (Office Math Paragraph Properties) 3679
22.1.2.80 opEmu (Operator Emulator) 3680
22.1.2.81 phant (Phantom Object) 3681
22.1.2.82 phantPr (Phantom Properties) 3682
22.1.2.83 plcHide (Hide Placeholders (Matrix)) 3682
22.1.2.84 pos (Position) 3683
22.1.2.85 postSp (Post-Paragraph Spacing) 3684
22.1.2.86 preSp (Pre-Paragraph Spacing) 3684
22.1.2.87 r (Run) 3685
22.1.2.88 rad (Radical Object) 3685
22.1.2.89 radPr (Radical Properties) 3686
22.1.2.90 rMargin (Right Margin) 3686
22.1.2.91 rPr (Run Properties) 3687
22.1.2.92 rSp (Row Spacing (Array)) 3688
22.1.2.93 rSpRule (Row Spacing Rule) 3689
22.1.2.94 scr (Script) 3691
22.1.2.95 sepChr (Delimiter Separator Character) 3692
22.1.2.96 show (Phantom Show) 3693
22.1.2.97 shp (Shape (Delimiters)) 3693
22.1.2.98 smallFrac (Small Fraction) 3694
22.1.2.99 sPre (Pre-Sub-Superscript Object) 3695
22.1.2.100 sPrePr (Pre-Sub-Superscript Properties) 3695
22.1.2.101 sSub (Subscript Object) 3696
22.1.2.102 sSubPr (Subscript Properties) 3697
22.1.2.103 sSubSup (Sub-Superscript Object) 3697
22.1.2.104 sSubSupPr (Sub-Superscript Properties) 3698
22.1.2.105 sSup (Superscript Object) 3699
22.1.2.106 sSupPr (Superscript Properties) 3699
22.1.2.107 strikeBLTR (Border Box Strikethrough Bottom-Left to Top-Right) 3700
22.1.2.108 strikeH (Border Box Strikethrough Horizontal) 3701
22.1.2.109 strikeTLBR (Border Box Strikethrough Top-Left to Bottom-Right) 3701
22.1.2.110 strikeV (Border Box Strikethrough Vertical) 3702
22.1.2.111 sty (style) 3703
22.1.2.112 sub (Subscript (Pre-Sub-Superscript)) 3704
22.1.2.113 subHide (Hide Subscript (n-ary)) 3704
22.1.2.114 sup (Superscript (Superscript object)) 3705
22.1.2.115 supHide (Hide Superscript (n-ary)) 3706
22.1.2.116 t (Text) 3707
22.1.2.117 transp (Transparent (Phantom)) 3707
22.1.2.118 type (Fraction type) 3708
22.1.2.119 vertJc (Vertical Justification) 3709
22.1.2.120 wrapIndent (Wrap Indent) 3711
22.1.2.121 wrapRight (Wrap Right) 3711

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.1.2.122 zeroAsc (Phantom Zero Ascent) 3712
22.1.2.123 zeroDesc (Phantom Zero Descent) 3713
22.1.2.124 zeroWid (Phantom Zero Width) 3714

22.1.3 Simple Types 3715

22.1.3.1 ST_BreakBin (Break Binary Operators) 3715
22.1.3.2 ST_BreakBinSub (Break on Binary Subtraction) 3715
22.1.3.3 ST_Char (Character) 3716
22.1.3.4 ST_FType (Fraction Type) 3716
22.1.3.5 ST_Integer2 (Integer value (-2 to 2)) 3717
22.1.3.6 ST_Integer255 (Integer value (1 to 255)) 3717
22.1.3.7 ST_Jc (Justification) 3717
22.1.3.8 ST_LimLoc (Limit Location) 3718
22.1.3.9 ST_Script (Script) 3718
22.1.3.10 ST_Shp (Shape (Delimiters)) 3719
22.1.3.11 ST_SpacingRule (Spacing Rule) 3719
22.1.3.12 ST_Style (Style) 3719
22.1.3.13 ST_TopBot (Top-Bottom) 3720
22.1.3.14 ST_UnSignedInteger (Unsigned integer.) 3720

End of informative text.

22.1.2 Elements

The following elements describe the contents of mathematical text.

22.1.2.1 acc (Accent)

This element specifies the accent function, consisting of a base and a combining diacritical mark. If accPr is omitted, the default accent is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*: Example accent functions are $\dot{a}$, $\acute{a}$, and $\tilde{a}$.

```
<m:acc>
  <m:accPr>
    <m:chr m:val="&#771;"/>
    <m:ctrlPr/>
  </m:accPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:acc>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Acc) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.2 accPr (Accent Properties)

This element specifies the properties of the Accent function. If chr is omitted, the default accent character is U+0302 (COMBINING CIRCUMFLEX ACCENT). [*Example*:

The diacritical mark ~ (tilde) is:

```
<m:accPr>
  <m:chr m:val="&#771;"/>
</m:accPr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AccPr) is located in §A.6.1. *end note*]

22.1.2.3 aln (Alignment)

This element specifies the alignment property on the box object. It is utilized only when the box is designated as an operator emulator. When 1 or true, this operator emulator serves as an alignment point; that is, designated alignment points in other equations can be aligned with it.

When the element is absent, the default is for the parent structure to not be an alignment point.

When the element is present and the val attribute is absent, the default value of the val attribute is 1 meaning that this property's parent structure is an alignment point.

[*Example*: For example, the following equation uses the operator emulator as an alignment point: $a == b$.

Its XML representation is as follows:

```
<m:oMath>
  <m:r>
    <m:t>a</m:t>
  </m:r>
  <m:box>
    <m:boxPr>
      <m:opEmu m:val="1"/>
      <m:aln m:val="1"/>
      <m:ctrlPr/>
    </m:boxPr>
    <m:e>
      <m:r>
        <m:t>==</m:t>
      </m:r>
    </m:e>
  </m:box>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:oMath>
```

end example]

[*Example*:

Given the following equations:

$$a + b - c = 2$$
$$x - y + z = 3$$

Their associated XML represenation is as follows:

```
<m:oMathPara>
  <m:oMath>
    <m:r>
      <m:t>a+b</m:t>
    </m:r>
    <m:r>
      <m:rPr>
        <m:aln/>
      </m:rPr>
      <m:t>-c=2</m:t>
    </m:r>
    <m:r>
      <m:rPr>
        <m:sty m:val="p"/>
      </m:rPr>
      <w:br/>
    </m:r>
  </m:oMath>
  <m:oMath>
    <m:r>
      <m:t>x</m:t>
    </m:r>
    <m:r>
      <m:rPr>
        <m:aln/>
      </m:rPr>
      <m:t>-y+z=3</m:t>
    </m:r>
  </m:oMath>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
</m:oMathPara>
```

The subtraction symbols in each of the above equations have been identified as alignment points, so the equations are aligned at their subtraction symbols (because they are in the same Math paragraph).

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.4 alnScr (Align Scripts)

This element specifies the alignment of scripts in the subscript/superscript function. When `1` or `true`, subscripts and superscripts are aligned to each other. When `0` or `false`, they are kerned to the shape of the base. If this element is omitted, scripts are not aligned. In other words, when the element is absent, the default is for the sub-superscript object to not align the superscript and subscript with each other.

When the element is present and the val attribute is absent, the default value of the val attribute is 1 meaning that this property is applied.

[*Example*: Example (OFF): f_2^3; Example (ON): f_2^3.

The XML representation of the second example above is:

```
<m:sSubSup>
  <m:sSubSupPr>
    <m:alnScr   m:val="1"/>
  </m:sSubSupPr>
  <m:e>
    <m:r>
      <m:t>f</m:t>
    </m:r>
  </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r>
    </m:sup>
  </m:sSubSup>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.1.2.5 argPr (Argument Properties)

This element specifies any properties of the math argument. [*Example*: The XML below represents the argSz attribute on the base element e of a box:

©ISO/IEC 2016 – All rights reserved

```
<m:box>
  <m:boxPr>
    <m:noBreak m:val="0"/>
    <m:ctrlPr/>
  </m:boxPr>
  <m:e>
    <m:argPr>
      <m:argSz m:val="-1"/>
    </m:argPr>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:box>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArgPr) is located in §A.6.1. *end note*]

22.1.2.6 argSz (Argument Size)

This element specifies the size, or script level, of an argument. If the element is omitted, the default argument size is 0.

Whether the element is absent or present without the val attribute, the default value of the val attribute is always `0`. However, this property does not specify an absolute size, rather it's absolute to it's starting size. Most arguments begin at the normal size which will be defined as `0` in the list below, but some begin, by default smaller or larger than the normal size. For instance, `1` represents that it is one size larger than normal, `2` represents that it is two sizes larger than normal, `-1` represents that it is one size smaller than normal, and `-2` represents that it is two sizes smaller than normal. The list below indicates the default starting sizes for each argument, if it is not in this list, then the size cannot be changed on that argument.

- box base: 0
- group-character base: 0
- lower limit limit: -1
- upper limit limit: -1
- n-ary operator subscript: -1
- n-ary operator superscript: -1
- radical degree: -2
- pre-sub-superscript subscript: -1
- pre-sub-superscript superscript: -1
- subscript subscript: -1
- sub-supersript subscript: -1

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

- sub-supersript superscript: -1
- superscript superscript: -1

[*Example*: The following example contains three runs: regular mathematical text, a box object with the base at script size (val=-1) and a box object with the base at script-script size (val=-2).

$$abc^{abc\ abc}$$

The XML below shows argSz used in the middle box.

```
<m:box>
  <m:boxPr>
    <m:noBreak m:val="0"/>
  </m:boxPr>
  <m:e>
    <m:argPr>
      <m:argSz m:val="-1"/>
    </m:argPr>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:box>
```

Because the size is set to -1 on the degree argument, the degree argument is 1 size smaller normal.

end example]

Attributes	Description
val (Value)	Specifies a value between -2 and 2 for the property defined by the parent XML element. The positive or negative sign specifies in which direction to change argument size; the absolute value specifies by how much. The table below represents two cases in which argument size can be changed: superscripts and boxes. In the superscript object a^{b^c}, by default the term c has script-script size. Should the user wish for the c to be shown at script size, `val` should be set to +1 (that is, one size larger). Should the user wish for c to be shown at text size, `val` should be set to +2 (that is, two sizes larger).

`val` of c in a^{b^c}	Display
Default	a^{b^c}
+1	a^{b^c}
+2	a^{b^c}

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><th>val of abc</th><th>Display</th></tr><tr><td>Default</td><td>abc</td></tr><tr><td>-1</td><td>abc</td></tr><tr><td>-2</td><td>abc</td></tr></table> The possible values for this attribute are defined by the ST_Integer2 simple type (§22.1.3.5).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Integer2) is located in §A.6.1. *end note*]

22.1.2.7 bar (Bar)

This element specifies the bar function, consisting of a base argument and an overbar or underbar, as in $\overline{a}$ and $\underline{a}$.

[*Example*: The XML below demonstrates the overbar in use.

```
<m:bar>
  <m:barPr>
   <m:pos   m:val="top"/>
  </m:barPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:bar>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Bar) is located in §A.6.1. *end note*]

22.1.2.8 barPr (Bar Properties)

This element specifies properties of the bar function. If this element is omitted, the bar assumes its default location of top (the mathematical overbar). [*Example*:

$\underline{x}$ (x with an underbar) is represented by the following XML representation:

```
<m:bar>
  <m:barPr>
    <m:pos m:val="bot"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </m:barPr>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:bar>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BarPr) is located in §A.6.1. *end note*]

22.1.2.9 baseJc (Matrix Base Justification)

This element specifies the justification of the matrix. Text outside of the matrix can be aligned with the bottom, top, or center of a matrix object. If this element is omitted, the matrix assumes center justification. In other words, whether the element is absent or present without the val attribute, the default of the val attribute is center.

[*Example*: This matrix has center baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

This matrix has top baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

This matrix has bottom baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$

The XML below represents the matrix with top baseJC:

```
  <m:d>
    <m:dPr>
      <m:shp  m:val="match"/>
    </m:dPr>
    <m:e>
      <m:m>
        <m:mPr>
          <m:baseJc m:val="top"/>
          <m:mcs>
            <m:mc>
              <m:mcPr>

                <m:count m:val="2"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <m:mcJc  m:val="center"/>
      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
<m:mr>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty m:val="p"/>
      </m:rPr>
      <m:t>1</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>2</m:t>
    </m:r>
  </m:e>
</m:mr>
<m:mr>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr  m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>3</m:t>
    </m:r>
  </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>4</m:t>
            </m:r>
          </m:e>
        </m:mr>
        <m:mr>
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>5</m:t>
            </m:r>
          </m:e>
          <m:e>
            <m:r>
              <m:rPr>
                <m:scr  m:val="roman"/>
                <m:sty  m:val="p"/>
              </m:rPr>
              <m:t>6</m:t>
            </m:r>
          </m:e>
        </m:mr>
      </m:m>
    </m:e>
  </m:d>
```

Because the matrix base justification is top, the top row of the matrix is lined up with the baseline of the rest of the line.

end example]

Attributes	Description
val (Value)	Specifies the vertical justification parent element respect to surrounding text. Possible values are `top`, `bottom`, and `center`. [*Example*: The following examples illustrate baseJc on the matrix object m.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	This matrix has center baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ This matrix has top baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ This matrix has bottom baseJc: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ The possible values for this attribute are defined by the ST_YAlign simple type (§22.9.2.20).

[*Note*: The W3C XML Schema definition of this element's content model (CT_YAlign) is located in §A.6.1. *end note*]

22.1.2.10 begChr (Delimiter Beginning Character)

This element specifies the beginning, or opening, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets, and braces. If this element is omitted, the default begChr is '('. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0028 (LEFT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters:

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.11 borderBox (Border-Box Object)

This element specifies the Border Box object, consisting of a border drawn around an instance of mathematical text (such as a formula or equation), as in $\boxed{a^2 + b^2 = c^2}$. If borderBoxPr is omitted then the default behavior of borderBox is a rectangular border (as shown in the "abc" example below).

[*Example*: The following example shows the XML representation of the following Border Box: $\boxed{abc}$

```
<m:borderBox>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderBox) is located in §A.6.1. *end note*]

22.1.2.12 borderBoxPr (Border-Box Properties)

This element specifies the properties of the Border Box object, which dictate the types of lines that can be drawn as part of the border. [*Example*: $\boxed{a^2 + b^2 = c^2}$ (Diagonal Strikethrough from Top Left) and $\underline{a^2 + b^2 = c^2}$ (no left or right edges). *end example*]

[*Example*:

$\lfloor x$ (left and bottom edges) is represented by the following XML:

```
<m:borderBox>
  <m:borderBoxPr>
<m:hideTop m:val="1"/>
    <m:hideRight m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>x</m:t>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      </m:r>
    </m:e>
  </m:borderBox>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BorderBoxPr) is located in §A.6.1. *end note*]

22.1.2.13 box (Box Object)

This element specifies the box object, which is used to group components of an equation or other instance of mathematical text. A boxed object can (for example) serve as an operator emulator with or without an alignment point, serve as a line break point, have associated argSz, or be grouped such as not to allow line breaks within. If boxPr is omitted, all properties will be "false" by default.

[*Example*: The mathematical text $a == b$ uses a box around the double equal sign.

Its XML representation is as follows:

```
<m:r>
  <m:t>a</m:t>
</m:r>
<m:box>
  <m:boxPr>
    <m:opEmu    m:val="1"/>
    <m:aln/>
  </m:boxPr>
  <m:e>
      <m:r>
        <m:t>==</m:t>
      </m:r>
  </m:e>
</m:box>
<m:r>
  <m:t>b</m:t>
</m:r>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Box) is located in §A.6.1. *end note*]

22.1.2.14 boxPr (Box Properties)

This element specifies properties of the Box object, for example, whether the Box serves as operator emulator with or without an alignment point, serves as a line break point, or receives the correct spacing for the mathematical differential. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

$a\ x\ b$ is partly represented by the following XML:

```
<m:box>
  <m:boxPr>
    <m:opEmu m:val="1"/>
  </m:boxPr>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:box>
```

The x is set to be an operator emulator so it is given operator spacing around it.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_BoxPr) is located in §A.6.1. *end note*]

22.1.2.15 brk (Break)

This element specifies whether there is a line break at the start of a run, or at the start of the Box object, such that the line wraps at the start of the run or box object. These user-defined line breaks occur when the XML tag `<m:brk/>` is encountered and does not follow a mathematical "order of precedence". If this element is omitted, a manual break is not inserted. In other words, when the element is absent, the default is for the parent structure to not manually break onto the next line. When the element is present and the val attribute is absent, the default of the val attribute is `0` meaning that this property's parent structure manually breaks onto the next line and is aligned with the beginning of the previous line.

The line may happen to wrap at this point if the mathematical text exceeds the column width. OMML does not specify behaviour for any kind of automatic line breaking. Instead it delegates this functionality to the underlying rendering engine similarly to how it does with the choice of operator spacing and the determination of whether an operator is unary or binary.

[*Example*: The following example includes a manual user-defined line break at the operator emulator:

$$a \\ == b$$

```
<m:r>
  <m:t>a</m:t>
</m:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:box>
  <m:boxPr>
    <m:opEmu    m:val="1"/>
    <m:brk/>
  </m:boxPr>
  <m:e>
    <m:r>
      <m:t>==</m:t>
    </m:r>
  </m:e>
</m:box>
<m:r>
  <m:t>b</m:t>
</m:r>
```

There is a break on the == symbol so a manual line break occurs right before the == symbol.

end example]

Attributes	Description
alnAt (Index of Operator to Align To)	Specifies the index of the operator on the previous line of mathematical text which shall be used as the alignment point for the current line of mathematical text . A line can be aligned to any operator on the previous line; this attribute specifies exactly which operator shall be the target of that alignment in cases where there are multiple operators. If alnAt is omitted, then all runs (r tag) that follow a brk tag will align with the left margin of the first run of mathematical text. [*Example*: For example, consider the break in this instance of mathematical text: $a + b + c + d + e$ $+ f + g$ The second line could theoretically be aligned to any of the four operators in the previous line. Specifying an alnAt value of 3 for the second line resolves this ambiguity; the second line is aligned to the third operator in the previous line. *end example*] The possible values for this attribute are defined by the ST_Integer255 simple type (§22.1.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_ManualBreak) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.1.2.16 brkBin (Break on Binary Operators)

This element specifies how binary operators are treated when they coincide with a line break. If this element is omitted, the line break occurs before the binary operator. That is, the binary operator is the first element on the wrapped line.

Whether the element is absent or present without the val attribute, the default of the val attribute is before.

[*Example*: For example:

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

before

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

after

$$f(x) = a_{11} + a_{12} + \cdots + a_{nn}$$

repeat

The above examples demonstrate how such an equation would look depending on the value of the brkBin document-property. These actual examples do not rely on the brkBin property of this document to maintain their configurations. This would be impossible because the brkBin property can only be set to one value for an entire document at any given time. Here we are demonstrating all three.

The XML below demonstrates brkBin in use under mathPr:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies where to break on binary operators. Possible values are before, after, and repeat. The possible values for this attribute are defined by the ST_BreakBin simple type (§22.1.3.1).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_BreakBin) is located in §A.6.1. *end note*]

22.1.2.17 brkBinSub (Break on Binary Subtraction)

This element specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. If this element is omitted, the subtraction operator is repeated before and after the break.

Whether the element is absent or present without the val attribute, the default of the val attribute is --.

[*Example*: The XML below demonstrates brkBinSub in use under mathPr:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

[*Example*:

Given the following XML representation using brkBinSub:

```
<m:brkBinSub m:val="-+"/>
```

Because the document option for break on binary subtraction is set to -+, any breaks in Math paragraphs on subtraction operators will display a – on the line before the break and a + on the line after the break.

end example]

Attributes	Description
val (Value)	Specifies how the subtraction operator is treated when it coincides with a line break, when brkBin is set to repeat. Possible values are --, -+, and +-. The possible values for this attribute are defined by the ST_BreakBinSub simple type (§22.1.3.2).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

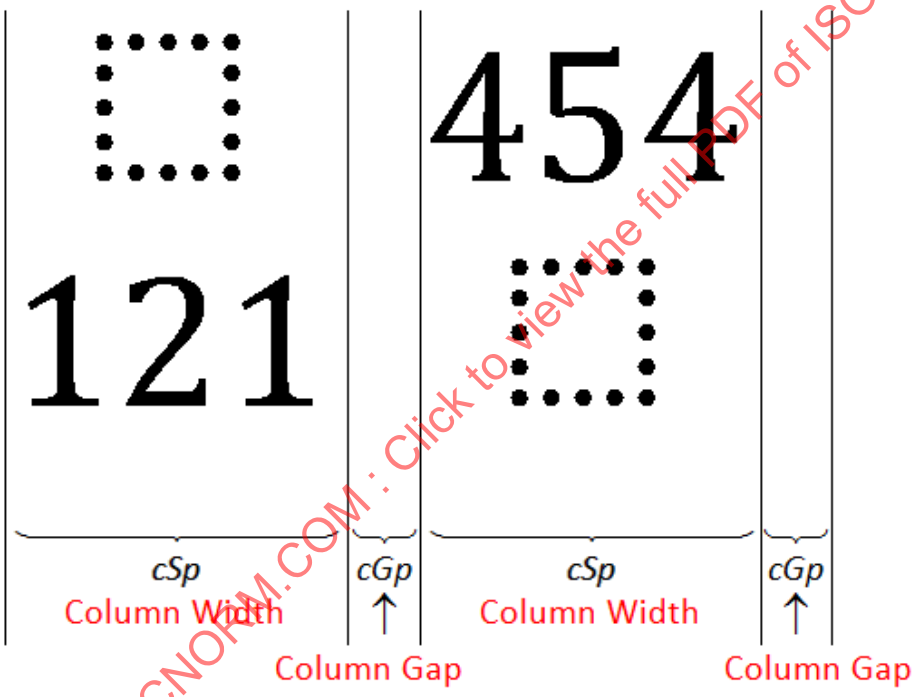
[*Note*: The W3C XML Schema definition of this element's content model (CT_BreakBinSub) is located in §A.6.1. *end note*]

22.1.2.18 cGp (Matrix Column Gap)

This element represents the (custom) column gap spacing information; the default value is `0` (which corresponds to 1 em). This value is interpreted differently depending on the value of cGpRule (§22.1.2.19). cGp is not used unless the value of cGpRule is 3 or 4. When cGpRule is omitted, the default spacing between matrix columns is 1 em (a val attribute value of `0`).

Whether the element is absent or present without the val attribute, the default of the val attribute is `0`.

The cGp gap spacing (also referred to as "Column Gap" or "Gap Width") is added to the cSp (Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The following image depicts how cGp and cSp work together to define matrix column spacing in a 2x2 matrix:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: With a cGpRule value of 4, the matrix: $\begin{matrix}1 & 2\\ 3 & 4\end{matrix}$ has 3 ems of spacing between columns (3 ems = 6 * 0.5 ems). The matrix properties that demonstrate this element in use are:

 ©ISO/IEC 2016 – All rights reserved

```
<m:mPr>
  <m:cGpRule  m:val="4"/>
  <m:cGp  m:val="6"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count m:val="2"/>
        <m:mcJc m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

See the documentation for cGpRule for more information about how cGp is interpreted for each value of cGpRule.

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments. For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.19 cGpRule (Matrix Column Gap Rule)

This element specifies the type of gap (horizontal spacing) between columns of a matrix; the default is 0. Horizontal spacing units can be ems or points (stored as twips).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single spacing gap).

[*Example*: The following matrix has double spacing (2 ems) between columns:

1 2
3 4

The XML that represents this property in use is:

```
<m:mPr>
  <m:cGpRule  m:val="2"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count  m:val="2"/>
        <m:mcJc  m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (Value)	Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:

Value	Column/Row Gap	Example
0	Single spacing gap (1 em)	1 2
1	1.5 spacing gap (1.5 ems)	1 2
2	2 spacing gap (2 ems)	1 2
3	Exactly (for columns, rely on value of cGp, measured in twips) (for rows, rely on value of rSp, measured in points)	1 2
4	Multiple (for columns, rely on value of cGp, measured in 0.5 em increments) (for rows, rely on value of rSp, measured in lines)	1 2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_SpacingRule simple type (§22.1.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SpacingRule) is located in §A.6.1. *end note*]

22.1.2.20 chr (Character)

This element specifies the character to be attached to the base of an accent object, a group character object, or an n-ary operator object. When the parent element is accPr, the chr value should be within the range of (U+0300–U+036F) or (U+20D0–U+20EF). When the parent element is groupChrPr, the chr value should be a horizontal stretch character, such as U+2190 (LEFTWARD ARROW). When the parent element is naryPr, the chr value should be an n-ary operator such as U+222B (INTEGRAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the corresponding objects character is absent.

If this property is omitted for accPr, the default accent character is U+0302 (ACCENT, COMBINING CIRCUMFLEX). If this property is omitted for groupChrPr, the default character is U+23DF (BOTTOM CURLY BRACKET). If this property is omitted for naryPr, the default character is U+222B (INTEGRAL).

In other words, when the element is absent, the defaults are as follows:

- accent object – Unicode character U+0302 (ACCENT, COMBINING CIRCUMFLEX)
- group-character object – Unicode character U+23DF (BOTTOM CURLY BRACKET)
- n-ary operator object – Unicode character U+222B (INTEGRAL)

[*Example*: Examples of accent characters are the dot, hat, and arrow in the following cases: $\dot{a}$ $\hat{a}$ $\vec{a}$.

For example, the following XML represents the acc $\tilde{a}$.

```
<m:acc>
  <m:accPr>
    <m:chr m:val="&#771;"/>
  </m:accPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:acc>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.21 count (Matrix Column Count)

This element specifies the number of columns to which a property applies.

Whether the element is absent or present without the val attribute, the default of the val attribute is 1.

[*Example*: The example below specifies that two of the columns in the matrix described by the XML have the `center` property.

```
<m:mPr>
  <m:cSp  m:val="120"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:count  m:val="2"/>
        <m:mcJc m:val="center"/>

      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (Value)	Specifies the number of columns to which a column property applies. [*Example*: A count attribute value of 3 specifies that the property applies to the next three columns of the matrix. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

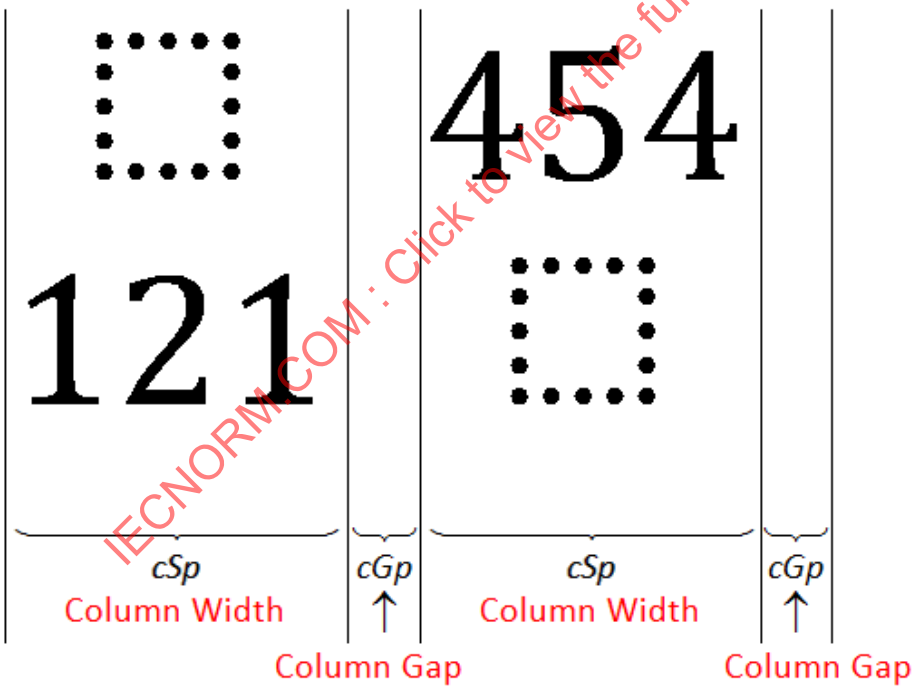
©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_Integer255 simple type (§22.1.3.6).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Integer255) is located in §A.6.1. *end note*]

22.1.2.22 cSp (Minimum Matrix Column Width)

This element specifies the minimum column width of a matrix. The actual column width of a matrix will be the greater of either the width of the column's widest argument or cSp. This additional spacing can be added to enhance appearance. If this element is omitted, the default minimum column width is 0. Whether the element is absent or present without the val attribute, the default of the val attribute is 0. The cGp gap spacing (also referred to as "Column Gap" or "Gap Width") is added to the cSp (Minimum Matrix Column Width) to determine the total Matrix Column Spacing (distance between the same edges of different columns). The value of cSp is interpreted as twips (a twip is 1/20th of a point). Therefore, a spacing of 1 point will be set by a cSp value of 20. This is the only use for cSp. There is no corresponding cSpRule. The following image depicts how cGp and cSp work together to define matrix column spacing in a 2x2 matrix:



IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: The following XML specifies that there should never be fewer than 6 pts (120 twips) between adjacent column edges of the matrix:

©ISO/IEC 2016 – All rights reserved

```
                                                    1  2
                                                    3  4
  <m:mPr>
    <m:cSp  m:val="120"/>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count  m:val="2"/>
          <m:mcJc m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
```

end example]

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments. For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.23 ctrlPr (Control Properties)

This element specifies properties on control characters; that is, object characters that cannot be selected. Examples of control characters are n-ary operators (excluding their limits and bases), fraction bars (excluding the numerator and denominator), and grouping characters (excluding the base). ctrlPr allows formatting properties to be stored on these control characters. The control character inherits its formatting from the paragraph

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

formatting; ctrlPr contains the formatting differences between the control character and the paragraph formatting.

If this element is omitted, the character properties of the first control character are the same as the first character in the math object.

[*Example*: The example below shows that the control character is of font Cambria Math. All other formatting, such as text size and color, are the same as the paragraph.

```
<m:ctrlPr>
  <w:rPr>
    <w:rFonts w:ascii="Cambria Math" w:hAnsi="Cambria Math"/>
  </w:rPr>
</m:ctrlPr>
```

end example]

CtrlPr is also used to save properties on characters used in the built-down form of an instance of mathematical text that are not displayed in Professional form. For example, the mathematical text (in linear form) ∫_0^1 might have color on the _ or ^. Though these characters are not displayed in Professional form, their formatting is stored such that their formatting will roundtrip through build up and build down.

[*Note*: The W3C XML Schema definition of this element's content model (CT_CtrlPr) is located in §A.6.1. *end note*]

22.1.2.24 d (Delimiter Object)

This element specifies the delimiter object, consisting of opening and closing delimiters (such as parentheses, braces, brackets, and vertical bars), and an element contained inside. The delimiter may have more than one element, with a designated separator character between each element. [*Example*:

Delimiter with one base: (x^2)

Delimiter with more than one base and separators, whose XML is shown below: $(x^2|y^2)$

```
<m:d>
  <m:e>
    <m:sSup>
      <m:e>
        <m:r>
          <m:t>x</m:t>
        </m:r>
      </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <m:sup>
          <m:r>
            <m:rPr>
              <m:scr  m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>2</m:t>
          </m:r>
        </m:sup>
      </m:sSup>
    </m:e>
    <m:e>
      <m:sSup>
        <m:e>
          <m:r>
            <m:t>y</m:t>
          </m:r>
        </m:e>
        <m:sup>
          <m:r>
            <m:rPr>
              <m:scr m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>2</m:t>
          </m:r>
        </m:sup>
      </m:sSup>
    </m:e>
  </m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_D) is located in §A.6.1. *end note*]

22.1.2.25 defJc (Default Justification)

This element specifies the default justification of display math, at the document level. Individual instances of mathematical text can overrule the default setting. If this element is omitted, the default justification is `centerGroup`. Whether the element is absent or present without the val attribute, the default of the val attribute is `centerGroup`.

Display math can be left justified, right justified, centered, or centered as a group. When display math is centered as a group, the mathematical text is left aligned within a block, and the entire block is centered with respect to column margins. If this element is omitted, the mathematical text is centered as a group.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: The XML below demonstrates defJc in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default justification of mathematical text in the document. Possible values are `center`, `centerGroup`, `left`, and `right`. The possible values for this attribute are defined by the ST_Jc simple type (§22.1.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathJc) is located in §A.6.1. *end note*]

22.1.2.26 deg (Degree)

This element specifies the degree in the mathematical radical. This element is optional. When omitted, the square root function, as in $\sqrt{x}$, is assumed. [*Example*:

The 3 in $\sqrt[3]{x}$ is represented by the XML below:

```
<m:rad>
  <m:deg>
     <m:r>
        <m:rPr>
          <m:scr   m:val="roman"/>
          <m:sty   m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r>
  </m:deg>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.27 degHide (Hide Degree)

This element specifies the per-object option to hide the degree of a radical. Every rad has a deg, but the deg can appear or not appear. When degHide is set to 1 or true, the degree is not shown, as in $\sqrt{x}$ (XML shown below). When degHide is omitted, the default is 0 or false; that is, the degree is not hidden. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*:

```
  <m:rad>
    <m:radPr>
      <m:degHide  m:val="1"/>
    </m:radPr>
    <m:deg>
    </m:deg>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.28 den (Denominator)

This element specifies the denominator of a fraction. [*Example*: For example, the b in $^{a}/_{b}$:

```
<m:f>
  <m:fPr>
    <m:type  m:val="skw"/>
  </m:fPr>
  <m:num>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:num>
  <m:den>
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:den>
</m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.29 diff (Differential)

The element specifies the differential property on box. When `1` or `true`, the box acts as a differential (e.g., dx in an integrand), and receives the appropriate horizontal spacing for the mathematical differential. When this property is omitted, the box is not treated as a differential.

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following example demonstrates a box set as differential in use, both in its proper form and in XML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

$$\int_0^1 x\,dx$$

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="&#8747;"/>
  </m:naryPr>
  <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>0</m:t>
      </m:r>
  </m:sub>
  <m:sup>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
    <m:box>
      <m:boxPr>
        <m:diff  m:val="1"/>
      </m:boxPr>
      <m:e>
        <m:r>
          <m:t>dx</m:t>
        </m:r>
      </m:e>
    </m:box>
  </m:e>
</m:nary>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.30 dispDef (Use Display Math Defaults)

This element specifies the document-level property to overwrite paragraph settings for mathematical text. When omitted, this element is set to `1` or `true` and special math settings are applied. Whether the element is absent or present without the val attribute, the default of the val attribute is `1` meaning that this option is applied.

[*Example*: The XML below demonstrates dispDef in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.31 dPr (Delimiter Properties)

This element specifies the properties of d, including the enclosing and separating characters, and the properties that affect the shape of the delimiters. [*Example*:

$[a + b]$ is represented by the following XML:

```
<m:d>
  <m:dPr>
    <m:begChr m:val="["/>
    <m:endChr m:val="]"/>
  </m:dPr>
  <m:e>
    <m:r>
      <m:t>a+b</m:t>
    </m:r>
  </m:e>
</m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DPr) is located in §A.6.1. *end note*]

22.1.2.32 e (Element (Argument))

This tag, which is an abbreviation for "element", serves several functions (18 total) including that of the base argument of a mathematical object or function, the elements in an array, and the elements in boxes. If all subelements are omitted, this element specifies the presence of an empty argument. [*Example*: The numerator of this function is an empty argument: $\frac{}{2}$. *end example*] [*Note*: The tag was shortened to e to improve readability over "element". "Element" was preferred over "Base" because "Element" sounded more neutral and appropriate to its multiple uses. *end note*]

The following table shows the 18 different uses for e:

Parent Element	Use
acc	Accent base argument

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Parent Element	Use
bar	Argument to which the bar is applied
borderBox	Argument around which the border box is drawn
box	Argument inside the abstract box
d	Argument inside the delimiters
eqArr	Each instance of mathematical text in the single-column array
func	Math argument list of the function
groupChr	Group character base
limLow	Base of the lower limit
limUpp	Base of the upper limit
mr	Each element in the matrix row
nary	n-ary and, e.g., integrand for an integral, summand for a summation
phant	Argument for the phantom
rad	Radicand
sPre	Base of the prescript object
sSub	Base of the subscript object
sSubSup	Base of the subsup object
sSup	Base of the superscript object

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*: For example, the func $\sin x$ has fName sin and e x:

```
<m:func>
  <m:fName>
    <m:r>
      <m:t>sin</m:t>
    </m:r>
  </m:fName>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

©ISO/IEC 2016 – All rights reserved

22.1.2.33 endChr (Delimiter Ending Character)

This element specifies the ending, or closing, delimiter character. Mathematical delimiters are enclosing characters such as parentheses, brackets, and braces. If this element is omitted, the default endChr is ')'. In other words, when the element is absent, the default is for the delimiter object beginning character to be Unicode character U+0029 (RIGHT PARENTHESIS).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object beginning character is absent.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters:

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

22.1.2.34 eqArr (Array Object)

This element specifies the Array object (sometimes referred to as "Equation Array", despite its ability to hold mathematical text other than equations), an object consisting of one or more equations, expressions, or other mathematical text runs that can be vertically justified as a unit with respect to surrounding text on the line. Alignment of multiple points within each run of mathematical text can occur within the array through the use of align values and spacer values. An *align value* is an ampersand within the array which acts as an alignment point (as described in §22.1.2.3). A *spacer value* is an ampersand (represented by "&" in the example below) within the array which designates where space can be added in order to align the align values on different rows of the array. Within each argument in the array, every odd ampersand is an align value and every even

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

ampersand is a spacer value (as well, the beginning of each argument provides an implied spacer value). If eqArrPr is omitted then the default values for its properties will be used. These defaults are:

Property	Default Value
baseJc	“center”
ctrlPr	<The character property of the first control character will be the character property of the first character in the eqArr object >
maxDist	“0”
objDist	“0”
rSp	“0”
rSpRule	“0” <single>

[*Example*:

A simple array, $\begin{matrix} a = b + c \\ d + e = f \end{matrix}$, is represented as follows:

```
<m:eqArr>
  <m:e>
    <m:r>
      <m:t>a=b+c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>d+e=f</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

A more complex array with alignment points, e.g.:

$$\begin{aligned} x - \ y + \ z &= \ 10 \\ 3x + \ y + 2z &= \ 34 \\ -5x + 2y - \ z &= -14 \end{aligned}$$

is represented as follows:

```
<m:eqArr>
    <m:e>
        <m:r>
            <m:t>&amp;x+&amp;&amp;y+&amp;&amp;z&amp;&amp;= &amp;&amp;10</m:t>
        </m:r>
    </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <m:e>
            <m:r>
                <m:t>3&amp;x+&amp;&amp;y+&amp;2&amp;z&amp;&amp;= &amp;&amp;34</m:t>
            </m:r>
        </m:e>
        <m:e>
            <m:r>
                <m:t>-5&amp;x+&amp;2&amp;y-&amp;&amp;z&amp;&amp;= &amp;&amp;-
14</m:t>
            </m:r>
        </m:e>
    </m:eqArr>
```

Notice that the variables, operators, and digits of the sums line up properly, as the align values line up between each argument by adding space at the position of each spacer value.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EqArr) is located in §A.6.1. *end note*]

22.1.2.35 eqArrPr (Array Properties)

This element specifies the properties of the array object, including the vertical justification of the object and layout inside the object. [*Example*:

$$\begin{aligned} 111a &= c \\ 111a + 111b &= 111c \\ b &= 111c \end{aligned}$$

```
<m:eqArr>
  <m:eqArrPr>
    <m:baseJc m:val="bottom"/>
  </m:eqArrPr>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;&amp;&amp;&amp;&amp;=&amp;&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;+&amp;111&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <m:t>&amp;amp;amp;amp;amp;amp;b&amp;amp;amp;=&amp;amp;amp;111&amp;amp;amp;c</m:t>
      </m:r>
    </m:e>
  </m:eqArr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_EqArrPr) is located in §A.6.1. *end note*]

22.1.2.36 f (Fraction Object)

This element specifies the fraction object, consisting of a numerator and denominator separated by a fraction bar. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar. [*Example*: Examples of fractions are:

Bar Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

No-Bar Fraction (Stack): $\binom{n}{k}$

The fraction a/b is represented as:

```
<m:f>
  <m:fPr>
    <m:type  m:val="skw"/>
  </m:fPr>
  <m:num>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:num>
  <m:den>
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:den>
</m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_F) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.1.2.37 fName (Function Name)

This element specifies the name of the function in the Function-Apply object func. For example, function names are sin and cos.

[*Example*: As an example, the func $\sin x$ has fName sin and e x:

```
<m:func>
  <m:fName>
    <m:r>
      <m:t>sin</m:t>
    </m:r>
  </m:fName>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
</m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.38 fPr (Fraction Properties)

This element specifies the properties of the fraction object f. Properties of the Fraction object include the type or style of the fraction. The fraction bar can be horizontal or diagonal, depending on the fraction properties. The fraction object is also used to represent the stack function, which places one element above another, with no fraction bar. [*Example*: Examples of fractions are:

Bar Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

No-Bar Fraction (Stack): $\binom{n}{k}$

The linear fraction c/d is represented by the following XML:

```
<m:f>
  <m:fPr>
    <m:type m:val="lin"/>
  </m:fPr>
  <m:num>
    <m:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <m:t>c</m:t>
      </m:r>
    </m:num>
    <m:den>
      <m:r>
        <m:t>d</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FPr) is located in §A.6.1. *end note*]

22.1.2.39 func (Function Apply Object)

This element specifies the Function-Apply object, which consists of a function name and an argument element (e) acted upon. It is often applied using a form of linear format. For example, in the linear format described in Unicode Technical Article #28, this object is applied by using the Function Application character (U+2061). [*Example*: Examples of Function-Apply objects include: $\sin x$, $\tan^{-1} x^2$, and $\max_{0 \le x \le 1} xe^{-x^2}$.

As an example, the func $\sin x$ has fName sin and e x:

```
  <m:func>
    <m:fName>
      <m:r>
        <m:t>sin</m:t>
      </m:r>
    </m:fName>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Func) is located in §A.6.1. *end note*]

22.1.2.40 funcPr (Function Properties)

This element specifies properties such as ctrlPr that can be stored on the function apply object func. [*Example*:

$tan\ x$ is represented by the following XML:

```
  <m:func>
    <m:funcPr/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:fName>
      <m:r>
        <m:rPr>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>tan</m:t>
      </m:r>
    </m:fName>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:func>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_FuncPr) is located in §A.6.1. *end note*]

22.1.2.41 groupChr (Group-Character Object)

This element specifies the Group-Character object, consisting of a character drawn above or below text, often with the purpose of visually grouping items. [*Example*: The following example demonstrates the groupChr in use, both in its proper form and in XML:

$$\overbrace{x+x+\cdots}$$

```
  <m:groupChr>
    <m:groupChrPr>
      <m:chr  m:val="&#9182;"/>
      <m:pos  m:val="top"/>
    </m:groupChrPr>
    <m:e>
      <m:r>
        <m:t>x+x+&#22EF;</m:t>
      </m:r>
    </m:e>
  </m:groupChr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupChr) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.42 groupChrPr (Group-Character Properties)

This element specifies the properties of the Group-Character object groupChr. These properties can be used to specify the character placed above or below the argument, and the position of the character. When omitted, character ⏟ (U+23DF, BOTTOM CURLY BRACKET) is used as the chr and its pos is set to bot.

[*Example*:

$\overbrace{x+y}$ is represented by the following XML:

```
<m:groupChr>
  <m:groupChrPr>
    <m:chr m:val="⏞"/>
    <m:pos m:val="top"/>
    <m:vertJc m:val="bot"/>
  </m:groupChrPr>
  <m:e>
    <m:r>
      <m:t>x+y</m:t>
    </m:r>
  </m:e>
</m:groupChr>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_GroupChrPr) is located in §A.6.1. *end note*]

22.1.2.43 grow (n-ary Grow)

This element specifies the growth property of n-ary operators. When `0` or `false`, n-ary operators such as integrals and summations do not grow to match the size of their operand height. When `1` or `true`, the n-ary operator grows vertically to match its operand height. If this property is omitted, grow is set to `0`.

When the element is absent, the default value of the property when it is a child of a delimiter object is `1` (meaning that this property is applied) and when it is a child of an n-ary operator object it is `0` (meaning that this property is not applied).

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The two integrals below demonstrate the difference between grow = `0` and grow = `1`.

$$\int_0^1 \frac{x^2}{x+y}dx \int_0^1 \frac{y^2}{x+y}dy$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The XML that defines nary growth is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:grow m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.44 hideBot (Hide Bottom Edge)

This element specifies the hidden or shown state of the bottom edge of borderBox. When this element is omitted, the bottom edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the bottom border is hidden, as in $\boxed{abc}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideBot/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.45 hideLeft (Hide Left Edge)

This element specifies the hidden or shown state of the left edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the left border is hidden, as in $\boxed{abc}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideLeft/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.46 hideRight (Hide Right Edge)

This element specifies the hidden or shown state of the right edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the right border is hidden, as in $\boxed{abc}$.

[*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideRight/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.47 hideTop (Hide Top Edge)

This element specifies the hidden or shown state of the top edge of borderBox. When this element is omitted, the edge is shown. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

When applied, the top border is hidden, as in $\boxed{abc}$. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.48 interSp (Inter-Equation Spacing)

This element specifies spacing between equations, expressions, or other instances of mathematical text within a display math paragraph, in twips.

©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.49 intLim (Integral Limit Locations)

This element specifies the document setting for the default placement of integral limits, when converted from a linear form to a two-dimensional output (professional form). Limits can be either centered above and below the integral, or positioned just to the right of the operator, as in:

$$\int_{a}^{b} x\,dx \quad \int_{a}^{b} x\,dx$$

When this integral object is written linearly, as ∫ _a^b, the placement of limits is ambiguous. intLim is a document-level property that specifies the default positioning. When this element is omitted, the default placement of integral limits is subSup (that is, the location of subscripts and superscripts, or just to the right of the base or operator). Whether the element is absent or present without the val attribute, the default of the val attribute is subSup.

[*Example*: The XML that specifies this property in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.50 intraSp (Intra-Equation Spacing)

This element specifies the spacing between adjacent display math paragraphs, in twips. If this element is omitted, no spacing is applied between adjacent math paragraphs.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.51 jc (Justification)

This element specifies justification of the math paragraph (a series of adjacent instances of mathematical text within the same paragraph). A math paragraph can be Left Justified, Right Justified, Centered, or Centered as Group. If this element is omitted, the math paragraph is Centered as Group. Whether the element is absent or present without the val attribute, the default of the val attribute is centerGroup. This means that the instances of mathematical text can be aligned with respect to each other, but the entire group of mathematical text is centered as a whole. [*Example*: An example of Centered as Group is the following example, in which each equation is left-aligned, but the series is centered:

$$x = x_1 + x_2 + x_3 + \cdots$$
$$y = y_1 + y_2 + y_3 + y_4 + \cdots$$
$$z = z_1 + z_2 + z_3 + z_4 + z_5 + \cdots$$

The XML that demonstrates jc in use is:

```
<m:oMathParaPr>
  <m:jc m:val="centerGroup"/>
</m:oMathParaPr>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Value)	Specifies the default justification of mathematical text in the document. Possible values are `center`, `centerGroup`, `left`, and `right`. The possible values for this attribute are defined by the ST_Jc simple type (§22.1.3.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathJc) is located in §A.6.1. *end note*]

22.1.2.52 lim (Limit)

This element specifies the lower limit of the limLow object and the upper limit of the limUpp function. [*Example*: For example, the limit of the limLow $\lim_{n\to\infty}$ is $n \to \infty$. The XML that specifies this object is:

```
<m:limLow>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:t>n&#8594;&#8734;</m:t>
    </m:r>
  </m:lim>
</m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.53 limLoc (n-ary Limit Location)

This element specifies the location of limits in n-ary operators. Limits can be either centered above and below the n-ary operator (shown in the first summation below), or positioned just to the right of the operator (shown in the second summation below).

$$\sum_{i=0}^{n} x_n \sum_{i=0}^{n} x_n$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

When the element is absent, the default value of this property is ignored and the intLim (for integral n-ary operator objects) and naryLim (for all other n-ary operator objects) options are looked to for the limit placement.

Ultimately, when this element is omitted, the default location is undOvr in display mode and subSup otherwise. When the element is present and the val attribute is absent, the default of the val attribute undOvr. [*Example*: The XML representing the relevant properties of the second summation above (subSup) is:

```
<m:naryPr>
  <m:chr  m:val="&#8721;"/>
  <m:limLoc m:val="subSup"/>
  <m:grow m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.54 limLow (Lower-Limit Object)

This element specifies the Lower-Limit object, consisting of text on the baseline and reduced-size text immediately below it. Examples of limLow include $\lim_{n\to\infty}$ and $\max_{0\le x\le 1}$. [*Example*: The XML that represents $\lim_{n\to\infty}$ is:

```
<m:limLow>
  <m:e>
    <m:r>
      <m:rPr>
        <m:scr m:val="roman"/>
        <m:sty  m:val="p"/>
      </m:rPr>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <m:lim>
      <m:r>
        <m:t>n&#8594;&#8734;</m:t>
      </m:r>
    </m:lim>
  </m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLow) is located in §A.6.1. *end note*]

22.1.2.55 limLowPr (Lower-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Lower Limit (limLow).

[*Example*:

$\lim_{a\to\infty}$ is represented by the following XML:

```
  <m:limLow>
    <m:limLowPr/>
    <m:e>
      <m:r>
        <m:t>lim</m:t>
      </m:r>
    </m:e>
    <m:lim>
      <m:r>
        <m:t>a→∞</m:t>
      </m:r>
    </m:lim>
  </m:limLow>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLowPr) is located in §A.6.1. *end note*]

22.1.2.56 limUpp (Upper-Limit Object)

This element specifies the Upper-Limit object, consisting of text on the baseline and reduced-size text immediately above it. [*Example*: Examples of limUpp include $\overbrace{x+x+x}^{k \text{ times}}$ and $\overset{\text{def}}{=}$.

The XML that specifies the $\text{limUpp}\overset{\text{def}}{=}$ is:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:limUpp>
  <m:e>
    <m:r>
      <m:t>=</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:rPr>
        <m:nor/>
      </m:rPr>
      <m:t>def</m:t>
    </m:r>
  </m:lim>
</m:limUpp>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimUpp) is located in §A.6.1. *end note*]

22.1.2.57 limUppPr (Upper-Limit Properties)

This element specifies control properties (ctrlPr) that can be stored on the Upper Limit (limUpp).

[*Example*:

$\lim_{a \to \infty}$ is represented by the following XML:

```
<m:limUpp>
  <m:limUppPr/>
  <m:e>
    <m:r>
      <m:t>lim</m:t>
    </m:r>
  </m:e>
  <m:lim>
    <m:r>
      <m:t>a→∞</m:t>
    </m:r>
  </m:lim>
</m:limUpp>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimUppPr) is located in §A.6.1. *end note*]

22.1.2.58 lit (Literal)

This element specifies that the characters in the run are literal; that is, they are to be interpreted literally and not be built up based on any implied mathematical meaning. This is especially useful for operators or other special characters that signal a need for build up to an OMML reader. These characters are often encountered during a given instance of mathematical text when presented in a 1-dimensional linear format, such as the linear format defined by Unicode Technical Note #28 (Sargent 2006).

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: In the following XML, the + operator is treated literally and does not receive proper binary spacing:

```
<m:r>a</m:r>
<m:r>
  <m:rPr>
    <m:lit/>
  </m:rPr>
  <m:t>+</m:t>
</m:r>
<m:r>
  <m:t>b</m:t>
</m:r>
```

As another example, consider the following Office Open XML Math markup:

```
<m:r>
    <m:t>a</m:t>
</m:r>
<m:r>
    <m:rPr>
        <m:lit/>
    </m:rPr>
    <m:t>/</m:t>
</m:r>
<m:r>
    <m:t>b</m:t>
</m:r>
```

This would be displayed as: a/b. If built up, it would normally be converted to $\frac{a}{b}$; however, the presence of this element specifies that the solidus should be treated literally, resulting in a/b.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.59 lMargin (Left Margin)

This element specifies the left margin for math, in twips. If this element is omitted, no left margin is used. In other words, when the element is absent, the default value of the option is `0`.

When the element is present and the val attribute is absent, the default of the val attribute is `1440` (or 1 inch).

Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored. [*Example*: The following XML demonstrates an lMargin setting of 1".

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="1440"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.60 m (Matrix Object)

This element specifies the Matrix object, consisting of one or more elements laid out in one or more rows and one or more columns. It is important to note that matrices do not have built in delimiters. Like other math elements, matrices are contained in a delimiter object (§22.1.2.24) when delimiters are desired. Empty arguments (see §22.1.2.32) can be used to create gaps in matrices. The plcHide tag (§22.1.2.83) can be used to indicate whether the empty arguments should be visible in the matrix (see the plcHide documentation for more information). If mPr is omitted, the values of baseJc, cGp, cGpRule, cSp, ctrlPr, mcs, plcHide, rSp, and rSpRule are shown in the following table:

Property	Default Value
baseJc	"center"
cGp	"0"
cGpRule	"0" <single>
cSp	"0"
ctrlPr	<The character property of the first control character shall be the character property of the first character in the m object >
mcs	<All columns will be vertically center aligned>
plcHide	"0"
rSp	"0"
rSpRule	"0" <single>

[*Example*: Examples of matrices are: $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$ and $\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$. Below is an example of a 2x2 matrix and its XML in its proper form (surrounding delimiters are not a property of the matrix and are only shown to demonstrate their relationship with a matrix. See the documentation for the d tag for XML examples of delimiters).

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:d>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
      </m:r >
    </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <m:e>
          <m:r>
            <m:rPr>
              <m:scr  m:val="roman"/>
              <m:sty  m:val="p"/>
            </m:rPr>
            <m:t>4</m:t>
          </m:r >
        </m:e>
      </m:mr>
    </m:m>
  </m:d>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_M) is located in §A.6.1. *end note*]

22.1.2.61 mathFont (Math Font)

This element specifies the default math font to be used in the document. If this element is omitted, font substitution (§17.8.2) should be used to determine the most appropriate font for use throughout the document. [*Example*: The XML containing this property is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="1440"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

[*Example*:

Given the following equation:

$a + b = c$

and the following XML:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:mathFont m:val="Cambria Math"/>
```

Because the document option for Math font is set to the mathematically enabled font Cambria Math, Math is formatted with that font.

end example]

Attributes	Description
val (value)	Specifies the default math font to be used in the document. The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_String) is located in §A.6.1. *end note*]

22.1.2.62 mathPr (Math Properties)

This element specifies the document-level properties for all math in the document.

[*Example*:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

These are the default settings for the math properties element and its children.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MathPr) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

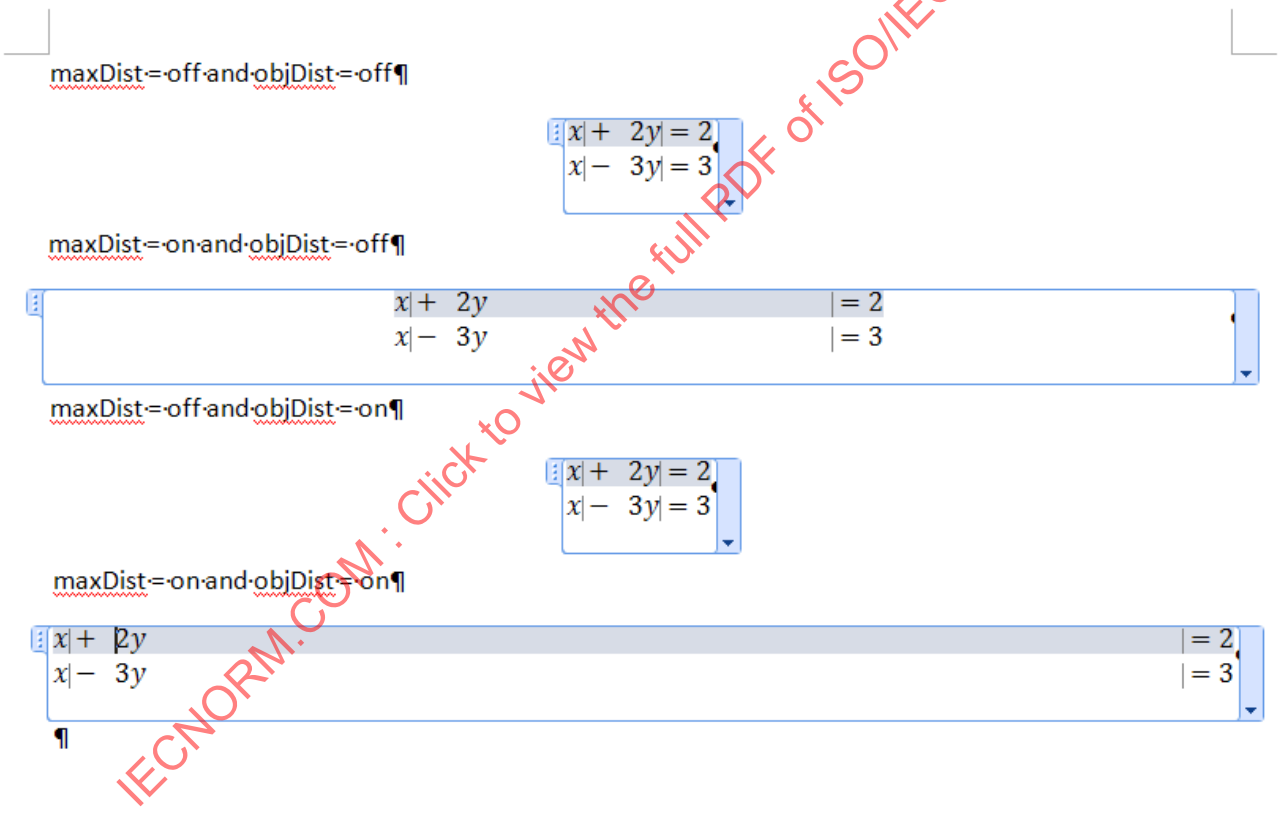
22.1.2.63 maxDist (Maximum Distribution)

This element specifies Array Maximum Distribution. When `1` or `true`, the array is spaced to the maximum width of the containing element (page, column, cell, etc.). The example image below illustrates an array expanded to fit the page, which is the containing element in this example. The maxDist option is commonly used with the objDist option. The objDist option is used to expand the distribution of mathematical text within the bounds of an array while not impacting the Array Distribution itself.

When the maxDist element is omitted, Array Maximum Distribution is off. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The following diagram illustrates all four possible combinations of maxDist and objDist:



XML for the fourth example shown in the above image is:

```
<m:oMathPara>
  <m:oMath>
    <m:eqArr>
      <m:eqArrPr>
        <m:maxDist m:val="1" />
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <m:objDist m:val="1" />
      </m:eqArrPr>
      <m:e>
        <m:r>
          <m:t>x</m:t>
        </m:r>
        <m:r>
           <m:t>&+</m:t>
        </m:r>
        <m:r>
          <m:t>2y&=2</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>x</m:t>
        </m:r>
        <m:r>
          <m:t>&-</m:t>
        </m:r>
        <m:r>
          <m:t>3y&=3</m:t>
        </m:r>
      </m:e>
    </m:eqArr>
  </m:oMath>
</m:oMathPara>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.1.2.64 mc (Matrix Column)

This element specifies a single column in a matrix m. [*Example*: An example of this element in use is:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MC) is located in §A.6.1. *end note*]

22.1.2.65 mcJc (Matrix Column Justification)

This element specifies the justification of a matrix column (or group of matrix columns) mc. When this element is omitted, the column is centered. Whether the element is absent or present without the val attribute, the default of the val attribute is center. The matrix below has three columns. The leftmost column is left-justified, the rightmost column is right-justified, and the center column is centered: $\begin{pmatrix} 1 & 1 & 1 \\ 23 & 23 & 23 \\ 456 & 456 & 456 \end{pmatrix}$

[*Example*: A simple example of this property in use is a 2x2 matrix with both columns centered:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
  <m:m>
    <m:mPr>
      <m:mcs>
        <m:mc>
          <m:mcPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <m:count m:val="2"/>
            <m:mcJc  m:val="center"/>

          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>2</m:t>
        </m:r >
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

Attributes	Description

val (Value)	Specifies the horizontal alignment of the parent element. Possible values are `left`, `right`, and `center`. [*Example*:
	`<m:mcPr>`
	`  <m:mcJc  m:val="center"/>`
	`  <m:count m:val="2"/>`
	`</m:mcPr>`
	
	The possible values for this attribute are defined by the ST_XAlign simple type (§22.9.2.18).

[*Note*: The W3C XML Schema definition of this element's content model (CT_XAlign) is located in §A.6.1. *end note*]

22.1.2.66 mcPr (Matrix Column Properties)

This element specifies the properties of the matrix column mn, including the number of columns and the type of justification. [*Example*: As an extreme example, the following matrix has two columns that are left justified (count is 2) and three columns that are right justified (count is 3). $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 23 & 23 & 23 & 23 & 23 \\ 456 & 456 & 456 & 456 & 456 \end{pmatrix}$ *end example*]

[*Example*:

$\begin{matrix} 1 & 2222 & 3 \\ 4444 & 5 & 6666 \end{matrix}$ is represented by the following XML:

```
  <m:m>
    <m:mPr>
      <m:mcs>
        <m:mc>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
          <m:mcPr>
            <m:count m:val="3"/>
            <m:mcJc m:val="right"/>
          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>1</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>2222</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>3</m:t>
        </m:r>
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>4444</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>5</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>6666</m:t>
        </m:r>
      </m:e>
    </m:mr>
  </m:m>
```

All three columns are right aligned.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MCPr) is located in §A.6.1. *end note*]

22.1.2.67 mcs (Matrix Columns)

This element specifies the collection of columns of the matrix m. [*Example*: An example of this element in use is:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
          <m:count m:val="2"/>
          <m:mcJc  m:val="center"/>

        </m:mcPr>
      </m:mc>
    </m:mcs>
  </m:mPr>
  <m:mr>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>1</m:t>
      </m:r >
    </m:e>
    <m:e>
      <m:r>
        <m:rPr>
          <m:scr  m:val="roman"/>
          <m:sty  m:val="p"/>
        </m:rPr>
        <m:t>2</m:t>
      </m:r >
    </m:e>
  </m:mr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MCS) is located in §A.6.1. *end note*]

22.1.2.68 mPr (Matrix Properties)

This element specifies properties of the matrix m, including the justification of the matrix and the layout of elements within the matrix. [*Example*:

$\begin{matrix} a+1 & 2+b \\ 3 & 4 \end{matrix}$ is represented by the following XML:

```
  <m:oMath>
    <m:r>
      <m:t>a+</m:t>
    </m:r>
    <m:m>
      <m:mPr>
        <m:baseJc m:val="top"/>
      </m:mPr>
      <m:mr>
        <m:e>
          <m:r>
            <m:t>1</m:t>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
          </m:r>
        </m:e>
        <m:e>
          <m:r>
            <m:t>2</m:t>
          </m:r>
        </m:e>
      </m:mr>
      <m:mr>
        <m:e>
          <m:r>
            <m:t>3</m:t>
          </m:r>
        </m:e>
        <m:e>
          <m:r>
            <m:t>4</m:t>
          </m:r>
        </m:e>
      </m:mr>
    </m:m>
    <m:r>
      <m:t>+b</m:t>
    </m:r>
  </m:oMath>
```

This top row of the matrix is aligned with the rest of the baseline.

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MPr) is located in §A.6.1. *end note*]

22.1.2.69 mr (Matrix Row)

This element specifies a single row of the matrix m. [*Example*: An example of this element in use is the following example, a 2x2 matrix. There are two rows; the first contains the elements 1 and 2; the second contains 3 and 4.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
<m:m>
  <m:mPr>
    <m:mcs>
      <m:mc>
        <m:mcPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <m:count m:val="2"/>
            <m:mcJc  m:val="center"/>

          </m:mcPr>
        </m:mc>
      </m:mcs>
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r >
      </m:e>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>2</m:t>
        </m:r >
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>3</m:t>
        </m:r >
      </m:e>
      <m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>4</m:t>
        </m:r >
      </m:e>
    </m:mr>
  </m:m>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_MR) is located in §A.6.1. *end note*]

22.1.2.70 nary (n-ary Operator Object)

This element specifies an n-ary object, consisting of an n-ary object, a base (or operand), and optional upper and lower limits. Examples of n-ary objects are: $\int_0^1 x\,dx$, $\sum_k \binom{n}{k}$, $\prod_{k=1}^{n} A_k$, and $\bigcup_{=1}^{m}(X_n \cap Y_n)$. [*Example*: The example below demonstrates an n-ary object in its proper form and XML representation:

$$\int_0^1 x\,dx$$

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="&#8747;"/>
  </m:naryPr>
  <m:sub>
      <m:r>
        <m:rPr>
          <m:scr m:val="roman"/>
          <m:sty m:val="p"/>
        </m:rPr>
        <m:t>0</m:t>
      </m:r>
  </m:sub>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:sup>
        <m:r>
          <m:rPr>
            <m:scr  m:val="roman"/>
            <m:sty  m:val="p"/>
          </m:rPr>
          <m:t>1</m:t>
        </m:r>
    </m:sup>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
      <m:box>
        <m:boxPr>
          <m:diff  m:val="1"/>
        </m:boxPr>
        <m:e>
          <m:r>
            <m:t>dx</m:t>
          </m:r>
        </m:e>
      </m:box>
    </m:e>
  </m:nary>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Nary) is located in §A.6.1. *end note*]

22.1.2.71 naryLim (n-ary Limit Location)

This element specifies the document setting for the default placement of n-ary limits other than integrals (since integrals are most often written as subSup and other n-ary operators are most often written as undOvr), when converted from a built down form to a two-dimensional output (professional form). Limits can be either centered above and below the n-ary operator, or positioned just to the right of the operator, as in:

$$\sum_{i=0}^{n} x_i \quad \sum_{i=0}^{n} x_i$$

When this summation object is written in built down form, as in ∑_(i = 0)^n, for example, the placement of limits is ambiguous. naryLim specifies this positioning. When this element is omitted, the default placement of n-ary limits is undOvr (that is, the location of lower and upper limits). Whether the element is absent or present without the val attribute, the default of the val attribute is undOvr.

[*Example*: An example XML of this element in use is:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the default location of limits on the parent object. Possible values are subSup and undOvr. The possible values for this attribute are defined by the ST_LimLoc simple type (§22.1.3.8).

[*Note*: The W3C XML Schema definition of this element's content model (CT_LimLoc) is located in §A.6.1. *end note*]

22.1.2.72 naryPr (n-ary Properties)

This element specifies the properties of the n-ary object.

These include the n-ary operator character that is used (using the chr element), the shape and height of the operator (using the grow element), the location of limits (using the limLoc element), and whether empty limits display a placeholder character or are hidden (using the subHide and supHide elements). [*Example*: N-ary operators include: $\sum$, $\int$, $\cup$, $\wedge$. *end example*]

[*Example*:

$\sum_{1}^{20} x$ is represented by the following XML:

```
<m:nary>
  <m:naryPr>
    <m:chr m:val="∑"/>
    <m:limLoc m:val="subSup"/>
  </m:naryPr>
  <m:sub>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
      <m:r>
        <m:t>1</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>20</m:t>
      </m:r>
    </m:sup>
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:nary>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NaryPr) is located in §A.6.1. *end note*]

22.1.2.73 noBreak (No Break)

This property specifies the "unbreakable" property on the Box object box. When `1` or `true`, no line breaks can occur within the box. This can be important for operator emulators that consist of more than one binary operator. When this element is not specified, breaks can occur inside box. Whether the element is absent or present without the val attribute, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: Sample XML containing this element is below. In this box, breaks are allowed.

```
  <m:boxPr>
    <m:noBreak m:val="0"/>
  </m:boxPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.74 nor (Normal Text)

This element specifies that the run is normal text, i.e., math italics and math spacing are not applied. In a normal text run, no characters will trigger reformatting of a linear expression into a two-dimensional expression.

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The example below illustrates three runs of normal text, along with the XML representation of the formula:

$$\text{rate} = \frac{\text{distance}}{\text{time}}$$

```
<m:r>
  <m:rPr>
    <m:nor/>
  </m:rPr>
  <m:t>rate</m:t>
</m:r>
<m:r>
  <m:t>=</m:t>
</m:r>
<m:f>
  <m:num>
    <m:r>
      <m:rPr>
        <m:nor/>
      </m:rPr>
      <m:t>distance</m:t>
    </m:r>
  </m:num>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <m:den>
      <m:r>
        <m:rPr>
          <m:nor/>
        </m:rPr>
        <m:t>time</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.75 num (Numerator)

This element specifies the numerator of the Fraction object f. [*Example*: The a in $^{a}/_{b}$:

```
<m:f>
  <m:fPr>
    <m:type  m:val="skw"/>
  </m:fPr>
  <m:num>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:num>
  <m:den>
    <m:r>
      <m:t>b</m:t>
    </m:r>
  </m:den>
</m:f>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.76 objDist (Object Distribution)

This element specifies Array Object Distribution. When `1` or `true`, the contents of the array are spaced to the maximum width of the array object. When this element is omitted, the array does not receive object distribution. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*:

$$\begin{array}{lllll} 111a & & & = & c \\ 111a & + & 111b & = & 111c \\ & & b & = & 111c \end{array}$$

is represented by the following XML:

```
<m:eqArr>
  <m:eqArrPr>
<m:maxDist m:val="1"/>
    <m:objDist m:val="1"/>
  </m:eqArrPr>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;&amp;&amp;&amp;&amp;=&amp;&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>111&amp;a&amp;&amp;+&amp;111&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>&amp;&amp;&amp;&amp;&amp;b&amp;&amp;=&amp;111&amp;c</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Because object distribution and maximum distribution are applied, the equation array is expanded to fill the width that is allowed and the columns are distributed horizontally within the object to the bounds of the object.

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.77 oMath (Office Math)

This element specifies an instance of mathematical text. When used independently (not inside an oMathPara) with non-mathematical text preceding and/or following it, an independent oMath is interpreted as an inline math zone. All such math zones, including equations, expressions, arrays of equations or expressions, and formulas are represented by oMath blocks. When used in a display math zone (a math paragraph, oMathPara), oMath is a container for an instance of mathematical text that starts on its own line and is not an inline math zone. When an oMath block is part of a display math zone, it is not itself an inline math zone. When an oMath block is not part of a display math zone, it is interpreted as its own inline math zone. The contents of an oMath block do not differ between display zone containers and independent inline math zones.

[*Example*:

$1 + 1 = 2$

```
<m:oMath>
  <m:r>
    <m:t>1+1=2</m:t>
  </m:r>
</m:oMath>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMath) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.78 oMathPara (Office Math Paragraph)

This element specifies a math paragraph, or display math zone, that contains one or more oMath elements that are in display mode. The oMath containers of a display math zone are not themselves considered inline math zones..

[*Example*:

$1 + 1 = 2$
$2 + 2 = 4$ is represented by the following XML:

```
<m:oMathPara>
  <m:oMath>
    <m:r>
      <m:t>1+1=2</m:t>
    </m:r>
    <m:r>
      <w:br/>
    </m:r>
  </m:oMath>
  <m:oMath>
    <m:r>
      <m:t>2+2=4</m:t>
    </m:r>
  </m:oMath>
</m:oMathPara>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathPara) is located in §A.6.1. *end note*]

22.1.2.79 oMathParaPr (Office Math Paragraph Properties)

This property specifies properties of the math paragraph oMathPara, including justification jc.

[*Example*:

$1 + 1 = 2$
$2 + 2 = 4$ is represented by the following XML:

```
<m:oMathPara>
  <m:oMathParaPr>
    <m:jc m:val="right"/>
  </m:oMathParaPr>
  <m:oMath>
    <m:r>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <m:t>1+1=2</m:t>
      </m:r>
      <m:r>
        <w:br/>
      </m:r>
    </m:oMath>
    <m:oMath>
      <m:r>
        <m:t>2+2=4</m:t>
      </m:r>
    </m:oMath>
  </m:oMathPara>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathParaPr) is located in §A.6.1. *end note*]

22.1.2.80 opEmu (Operator Emulator)

This element specifies the Operator Emulator property on box. When 1 or `true`, the box and its contents behave as a single operator and inherit the properties of an operator. This means, for example, that the character can serve as a point for a line break and can be aligned to other operators. (For more details on the properties of an operator, see Unicode Technical Report #25, §3.2.2 and §3.2.3, and Unicode Technical Note #28.) Operator Emulators are often used when one or more glyphs combine to form an operator, such as $==$.

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: The following equation uses an Operator Emulator:

$$a == b$$

Its XML representation is as follows:

```
<m:r>
  <m:t>a</m:t>
</m:r>
<m:box>
  <m:boxPr>
    <m:opEmu m:val="1"/>
    <m:aln/>
  </m:boxPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
      <m:e>
        <m:r>
          <m:t>==</m:t>
        </m:r>
      </m:e>
    </m:box>
    <m:r>
      <m:t>b</m:t>
    </m:r>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.81 phant (Phantom Object)

This element specifies the phantom object. This object has two primary uses: adding the spacing of the phantom base element e without displaying that base; and suppressing part of the glyphfor spacing considerations. [*Example*: In the right column below, the descender of the y is suppressed (set to zero) while in the left column, it is shown. The presence or absence of the descender changes the shape of the $\sqrt{y}$.

Without <m:phant> $\sqrt{y}$	With <m:phant> $\sqrt{y}$

`<m:rad>`
`  <m:radPr>`
`  <m:degHide m:val="1" />`
`  </m:radPr>`
`  <m:deg />`
`  <m:e>`
`  <m:r>`
`  <m:t>y</m:t>`
`  </m:r>`
`  </m:e>`
	`<m:rad>`
	`  <m:radPr>`
	`  <m:degHide m:val="1" />`
	`  </m:radPr>`
	`  <m:deg />`
	`  <m:e>`
	`  <m:phant>`
	`  <m:phantPr>`
	`  <m:zeroDesc m:val="1" />`
	`  </m:phantPr>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

</m:rad>	<m:e> <m:r> <m:t>y</m:t> </m:r> </m:e> </m:phant> </m:e> </m:rad>

end example]

[*Note*: In typography, a *glyph* is the shape given in a particular typeface to a specific symbol. Most scripts share the notion of a baseline: an imaginary horizontal line on which characters rest. In some scripts, parts of glyphs lie below the baseline. The *descent* spans the distance between the baseline and the lowest descending glyph in a typeface, and the part of a glyph that descends below the baseline has the name *descender*. Conversely, the *ascent* spans the distance between the baseline and the top of the glyph that reaches farthest from the baseline. end note].

[*Note*: The W3C XML Schema definition of this element's content model (CT_Phant) is located in §A.6.1. *end note*]

22.1.2.82 phantPr (Phantom Properties)

This element specifies properties of the Phantom object, including whether the phantom is hidden or visible, and the amount of space that is taken into account when laying out text and objects around phantoms.

[*Note*: The W3C XML Schema definition of this element's content model (CT_PhantPr) is located in §A.6.1. *end note*]

22.1.2.83 plcHide (Hide Placeholders (Matrix))

This element specifies the Hide Placeholders property on a matrix m. When this property is on, placeholders do not appear in the matrix. If this element is omitted, placeholders do appear such that the locations where text can be inserted are made visible. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: The following two examples of matrices show the hidden (right matrix) and visible (left matrix) states of placeholders. The key difference between the placeholders in these matrices is that the left matrix contains visual representations for each placeholder in the matrix while the right matrix does not. However, the contents of each matrix remain the same between these two examples:

$$\begin{pmatrix} 1 & \square & \square \\ \square & 1 & \square \\ \square & \square & 1 \end{pmatrix}\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}$$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

The plcHide element does not specify how a placeholder should be visually represented. It only specifies whether placeholders should or should not be visualized. The above example (left) is only one possible visual representation.

In the matrix described by the XML below, placeholders will be hidden:

```
<m:mPr>
  <m:plcHide  m:val="1"/>
  <m:mcs>
    <m:mc>
      <m:mcPr>
        <m:mcJc m:val="center"/>
        <m:count  m:val="3"/>
      </m:mcPr>
    </m:mc>
  </m:mcs>
</m:mPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.84 pos (Position)

This element specifies the position of the bar or group character in the parent object; the default is bot. Whether the element is absent or present without the val attribute, the default of the val attribute is bot.

[*Example*: The XML representation for the mathematical overbar is:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<m:bar>
  <m:barPr>
    <m:pos    m:val="top"/>
  </m:barPr>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:bar>
```

end example]

Attributes	Description

val (Value)	Specifies the position of the parent element. Possible values are `top` and `bot`.
	
	[*Example*:
	`<m:barPr>`
	`  <m:pos  m:val="top"/>`
	`</m:barPr>`
	
	The possible values for this attribute are defined by the ST_TopBot simple type (§22.1.3.13).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TopBot) is located in §A.6.1. *end note*]

22.1.2.85 postSp (Post-Paragraph Spacing)

This element specifies the spacing after a math paragraph, in twips. If this element is omitted, no spacing is applied after the paragraph.

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.86 preSp (Pre-Paragraph Spacing)

This element specifies the spacing before a math paragraph, in twips. If this element is omitted, no spacing is applied before the paragraph.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.87 r (Run)

This element specifies a run of mathematical text.

[*Example*:

$j + k$ is represented by the following XML:

```
<m:oMath>
  <m:e>
    <m:r>
      <m:t>j+k</m:t>
    </m:r>
  </m:e>
</m:oMath>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_R) is located in §A.6.1. *end note*]

22.1.2.88 rad (Radical Object)

This element specifies the radical object, consisting of a radical, a base e, and an optional degree deg. [*Example*: Examples of rad are $\sqrt[3]{x}$ (XML shown below) and $\sqrt{x}$.

```
<m:rad>
  <m:deg>
     <m:r>
       <m:rPr>
          <m:scr    m:val="roman"/>
          <m:sty    m:val="p"/>
        </m:rPr>
        <m:t>3</m:t>
     </m:r>
  </m:deg>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <m:e>
      <m:r>
        <m:t>x</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Rad) is located in §A.6.1. *end note*]

22.1.2.89 radPr (Radical Properties)

This element specifies properties of the Radical object rad, including the hidden or shown state of the degree deg. [*Example*:

$\sqrt{b}$ is represented by the following XML:

```
  <m:rad>
    <m:radPr>
      <m:degHide m:val="1"/>
    </m:radPr>
    <m:deg/>
    <m:e>
      <m:r>
        <m:t>b</m:t>
      </m:r>
    </m:e>
  </m:rad>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RadPr) is located in §A.6.1. *end note*]

22.1.2.90 rMargin (Right Margin)

This element specifies the right margin for math, in twips. If this element is omitted, no right margin is used. In other words, when the element is absent, the default value of the option is 0. When the element is present and the val attribute is absent, the default of the val attribute is 1440 (or 1 inch).Math margins are added to the paragraph settings for margins. If the sum of lMargin and rMargin exceed the width available, lMargin should be ignored. If rMargin exceeds the width available, a default indent of 1440 twips should be used. [*Example*: The following XML demonstrates an rmargin setting of 1".

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="1440"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.91 rPr (Run Properties)

This element specifies the properties of the math run r.

[*Example*:

$\mathfrak{F} + \mathfrak{G}$ is represented by the following XML:

```
<m:r>
  <m:rPr>
    <m:scr m:val="fraktur"/>
    <m:sty m:val="b"/>
  </m:rPr>
  <m:t>F+G</m:t>
</m:r>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_RPR) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.1.2.92 rSp (Row Spacing (Array))

This element specifies spacing between rows of an array eqArr; it is used only when rSpRule is set to 3 (exactly; in which case the unit of measure is points) or 4 (Multiple; in which case the unit of measure is half-lines). If this element is omitted, single line spacing is used in the array, and no additional spacing is used in the layout of rows. Whether the element is absent or present without the val attribute, the default of the val attribute is 0.

[*Example*: Below are three examples of the same array, with single line spacing, 1.5 line spacing, and double line spacing:

$$\begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & 34 \\ -5x+2y- & z= & -14 \end{pmatrix} \begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & 34 \\ -5x+2y- & z= & -14 \end{pmatrix} \begin{pmatrix} x- & y+ & z= & 10 \\ 3x+ & y+2z= & 34 \\ -5x+2y- & z= & -14 \end{pmatrix}$$

The following eqArr $\begin{matrix} a=b+c \\ d+e=f \end{matrix}$ has rSp of 3 half-lines:

```
<m:eqArr>
  <m:eqArrPr>
    <m:rSpRule  m:val="4"/>
    <m:rSp m:val="3"/>
  </m:eqArrPr>
  <m:e>
    <m:r>
      <m:t>a=b+c</m:t>
    </m:r>
  </m:e>
  <m:e>
    <m:r>
      <m:t>d+e=f</m:t>
    </m:r>
  </m:e>
</m:eqArr>
```

end example]

Attributes	Description
val (Value)	Specifies the amount of space between columns of the parent element (for cGp/cSp) or rows (for rSp). The manner in which this value is determined depends on the setting of the rule of the parent element. For cGp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as twips. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as number of 0.5 em increments.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	For cSp, this value is measured in twips. There is no corresponding cSpRule. For rSp, if the rule is set to 3 (or "Exactly"), then the unit is interpreted as points. If the rule is set to 4 (or "Multiple"), then the unit is interpreted as half-lines. The possible values for this attribute are defined by the ST_UnSignedInteger simple type (§22.1.3.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_UnSignedInteger) is located in §A.6.1. *end note*]

22.1.2.93 rSpRule (Row Spacing Rule)

This element specifies the type of vertical spacing between columns in a matrix. The following table demonstrates possible values of rSpRule along with their definitions and examples.

Value	Line spacing between rows	Example (non-normative)
0	Single line gap	1 2 3 4
1	1.5 line gap	1 2 3 4
2	2 line gap	1 2 3 4
3	Exactly (rely on value of rGp, measured in points)	1 2 3 4
4	Multiple (rely on value of rGp, measured in lines)	1 2 3 4

Whether the element is absent or present without the val attribute, the default of the val attribute is 0 (or single line gap).

[*Example*:

1 2
3 4
is represented by the following XML:

```
<m:m>
  <m:mPr>
    <m:rSpRule m:val="2"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </m:mPr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>1</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>2</m:t>
        </m:r>
      </m:e>
    </m:mr>
    <m:mr>
      <m:e>
        <m:r>
          <m:t>3</m:t>
        </m:r>
      </m:e>
      <m:e>
        <m:r>
          <m:t>4</m:t>
        </m:r>
      </m:e>
    </m:mr>
  </m:m>
```

Because the row spacing rule is set to 2, there is twice as much space between the matrix rows as usual.

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
val (Value)	Specifies the type of spacing between rows and/or columns. Possible values are 0, 1, 2, 3, or 4, whose definitions are contained in the following table:

Value	Column/Row Gap	Example
0	Single spacing gap (1 em)	1 2
1	1.5 spacing gap (1.5 ems)	1 2
2	2 spacing gap (2 ems)	1 2
3	Exactly (for columns, rely on value of cGp, measured in twips) (for rows, rely on value of	1 2

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	<table><tr><td></td><td>rSp, measured in points)</td><td></td></tr><tr><td>4</td><td>Multiple (for columns, rely on value of cGp, measured in 0.5 em increments) (for rows, rely on value of rSp, measured in lines)</td><td>1 2</td></tr></table> The possible values for this attribute are defined by the ST_SpacingRule simple type (§22.1.3.11).

[*Note*: The W3C XML Schema definition of this element's content model (CT_SpacingRule) is located in §A.6.1. *end note*]

22.1.2.94 scr (Script)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character (between U+0000 and U+007F), along with the script of the character. The application maps the value and script type to the appropriate Unicode range.

Whether the element is absent or present without the val attribute, the default of the val attribute is roman.

[*Example*: 𝔞𝕒

```
<m:r>
  <m:rPr>
    <m:scr m:val="fraktur"/>
    <m:sty m:val="p"/>
  </m:rPr>
  <m:t>a</m:t>
</m:r>
<m:r>
  <m:rPr>
    <m:scr m:val="double-struck"/>
    <m:sty m:val="p"/>
  </m:rPr>
  <m:t>a</m:t>
</m:r>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
val (Value)	Specifies the script type of the parent element. Possible values are: `double-struck`, `fraktur`, `monospace`, `roman`, `sans-serif`, and `script`. The possible values for this attribute are defined by the ST_Script simple type (§22.1.3.9).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Script) is located in §A.6.1. *end note*]

22.1.2.95 sepChr (Delimiter Separator Character)

This element specifies the character that separates base arguments e in the delimiter object d. If this element is omitted, the default sepChr is '|'. In other words, when the element is absent, the default is for the delimiter object separator character to be U+2502 (BOX DRAWINGS LIGHT VERTICAL).

When the element is present and the val attribute is absent, the default of the val attribute is empty which means the delimiter object separator character is absent.

[*Example*: Examples of d, each with a different sepChr, are: $(a_1|a_2)(a_1:a_2)(a_1;a_2)$. The following example sets COLON (:) as the separator character:

```
<m:dPr>
  <m:sepChr val="&#0058;"/>
</m:dPr>
```

end example]

Attributes	Description

val (value)	Specifies the character used by the parent element. When it is omitted, the parent uses its assigned default.
	
	[*Example*: Accent object ã:
	
	`<m:accPr>`
	`  <m:chr m:val="̃"/>`
	`</m:accPr>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Char simple type (§22.1.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Char) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.96 show (Phantom Show)

This element specifies the show property of the phantom phant. When `0` or `false`, the phant base e is hidden. If this element is omitted, the base e is shown. Whether the element is absent or present without the val attribute, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: In the following example, there is a phantom of the fraction a/b in the second radical such that only the height is preserved. The fraction does not show.

$$\sqrt{\frac{a}{b}} = \sqrt{x}$$

```
<m:phantPr>
  <m:show  m:val="0"/>
  <m:zeroDesc m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.97 shp (Shape (Delimiters))

This element specifies the shape of delimiters in the delimiter object d. Delimiters can be centered around the math axis of the mathematical text and still be made to fit the entire height of their contents (see right-hand example below), or their height and shape can be altered to exactly match their contents (see left-hand example below). These settings significantly impact the shape of the mathematical text. When this element is omitted, delimiters are 'centered'. Whether the element is absent or present without the val attribute, the default of the val attribute is centered. [*Example*: In the examples below, delimiters will be matched to the exact shape of their contents contents on the left, and will be centered on the right:

Match Centered

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

$$a+b=\left(\frac{\frac{a}{b}}{\frac{z}{r}}+\frac{42}{z+y}\right)$$

```
<m:dPr>
  <m:shp  m:val="match"/>
</m:dPr>
```

$$a+b=\left(\frac{\frac{a}{b}}{\frac{z}{r}}+\frac{42}{z+y}\right)$$

```
<m:dPr>
  <m:shp  m:val="centered"/>
</m:dPr>
```

end example]

Attributes	Description
val (Value)	Specifies the shape of the parent element. Possible values are `match` and `centered`. The possible values for this attribute are defined by the ST_Shp simple type (§22.1.3.10).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Shp) is located in §A.6.1. *end note*]

22.1.2.98 smallFrac (Small Fraction)

This element specifies a reduced fraction size display math, such that the numerator and denominator are written in script size instead of at the size of regular text.

When the element is absent, the default value of the option is `0` meaning that this option is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this option is applied.

[*Example*: The XML containing this element in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.99 sPre (Pre-Sub-Superscript Object)

This element specifies the Pre-Sub-Superscript object, which consists of a base e and a subscript and superscript placed to the left of the base, as in ${}_{1}^{2}A$. [*Example*: The XML that specifies this function is:

```
<m:sPre>
  <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>A</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SPre) is located in §A.6.1. *end note*]

22.1.2.100 sPrePr (Pre-Sub-Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Pre-Sub-Superscript object sPre.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Example*:

${}_{1}^{2}a$ is represented by the following XML:

```
<m:sPre>
  <m:sPrePr/>
  <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SPrePr) is located in §A.6.1. *end note*]

22.1.2.101 sSub (Subscript Object)

This element specifies the subscript object sSub, which consists of a base e and a reduced-size scr placed below and to the right, as in x_n. [*Example*: The XML that specifies this object is:

```
<m:sSub>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
  <m:sub>
    <m:r>
      <m:t>n</m:t>
    </m:r>
  </m:sub>
</m:sSub>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSub) is located in §A.6.1. *end note*]

22.1.2.102 sSubPr (Subscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Subscript object sSub.

[*Example*:

a_2 is represented by the following XML:

```
<m:sSub>
  <m:sSubPr/>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
  <m:sub>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sub>
</m:sSub>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubPr) is located in §A.6.1. *end note*]

22.1.2.103 sSubSup (Sub-Superscript Object)

This element specifies the sub-superscript object, which consists of a base e, a reduced-size scr placed below and to the right, and a reduced-size scr placed above and to the right, as in x_m^n.

[*Example*: The XML that specifies this function is:

```
<m:sSubSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </r>
  </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <m:sub>
      <m:r>
        <m:t>m</m:t>
      </r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>n</m:t>
      </r>
    </m:sup>
  </m:sSubSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubSup) is located in §A.6.1. *end note*]

22.1.2.104 sSubSupPr (Sub-Superscript Properties)

This element specifies properties of the Sub-Superscript object, including the alignment of scripts.

[*Example*:

f_{100}^{200} is represented by the following XML:

```
  <m:sSubSup>
    <m:sSubSupPr>
      <m:alnScr m:val="1"/>
    </m:sSubSupPr>
    <m:e>
      <m:r>
        <m:t>f</m:t>
      </m:r>
    </m:e>
    <m:sub>
      <m:r>
        <m:t>100</m:t>
      </m:r>
    </m:sub>
    <m:sup>
      <m:r>
        <m:t>200</m:t>
      </m:r>
    </m:sup>
  </m:sSubSup>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSubSupPr) is located in §A.6.1. *end note*]

22.1.2.105 sSup (Superscript Object)

This element specifies the superscript object sSup, which consists of a base e and a reduced-size scr placed above and to the right, as in x^n. [*Example*: The XML that specifies this object is:

```
<m:sSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </r>
  </m:e>
  <m:sup>
    <m:r>
      <m:t>n</m:t>
    </r>
  </m:sup>
</m:sSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSup) is located in §A.6.1. *end note*]

22.1.2.106 sSupPr (Superscript Properties)

This element specifies properties such as ctrlPr that can be stored on the Superscript object sSup.

[*Example*:

a^2 is represented by the following XML:

```
<m:sSup>
  <m:sSupPr/>
  <m:e>
    <m:r>
      <m:t>a</m:t>
    </m:r>
  </m:e>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

</m:sSup>*end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SSupPr) is located in §A.6.1. *end note*]

22.1.2.107 strikeBLTR (Border Box Strikethrough Bottom-Left to Top-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the bottom-left corner to the top-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.When applied, a strikethrough is drawn, as in ~~abc~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeBLTR m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.108 strikeH (Border Box Strikethrough Horizontal)

This element specifies the hidden or shown state of a strikethrough horizontal line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied. When on, a horizontal strikethrough is drawn, as in ~~*abc*~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeH m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.109 strikeTLBR (Border Box Strikethrough Top-Left to Bottom-Right)

This element specifies the hidden or shown state of a strikethrough diagonal line from the top-left corner to the bottom-right corner of borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

applied. When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied. When applied, a strikethrough is drawn, as in ~~abc~~. [*Example*:

```
<m:borderBox>
  <m:borderBoxPr>
    <m:hideTop  m:val="1"/>
    <m:hideBot  m:val="1"/>
    <m:hideLeft  m:val="1"/>
    <m:hideRight  m:val="1"/>
    <m:strikeTLBR m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.110 strikeV (Border Box Strikethrough Vertical)

This element specifies the hidden or shown state of a strikethrough vertical line in borderBox. When this element is omitted, the strikethrough is not drawn. In other words, when the element is absent, the default value of the property is 0 meaning that this property is not applied. When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied. When applied, a strikethrough is drawn, as in ϕ. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<m:borderBox>
  <m:borderBoxPr>
    <m:strikeV m:val="1"/>
  </m:borderBoxPr>
  <m:e>
    <m:r>
      <m:t>abc</m:t>
    </m:r>
  </m:e>
</m:borderBox>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.111 sty (style)

This element describes the script applied to the characters in the run. The XML includes the Unicode value of the character along with the style of the character. The application maps the value and style to the appropriate Unicode range. Whether the element is absent or present without the val attribute, the default of the val attribute is `i`.

[*Example*: **ab**

```
<m:oMath>
  <m:r>
    <m:rPr>
      <m:scr  m:val="roman"/>
      <m:sty m:val="b"/>
    </m:rPr>
    <m:t>ab</m:t>
  </m:r>
</m:oMath>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

end example]

Attributes	Description
val (Value)	Specifies the style of the parent element. Possible values are `b` (bold), `i` (italic), `bi` (bold-italic), and `p` (plain). The possible values for this attribute are defined by the ST_Style simple type (§22.1.3.12).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Style) is located in §A.6.1. *end note*]

22.1.2.112 sub (Subscript (Pre-Sub-Superscript))

This element specifies the subscript of the Pre-Sub-Superscript object sPre. [*Example*: For example, the sub in the object ${}_{1}^{2}A$ is 1. An example of this element in use is:

```
<m:sPre>
   <m:sub>
    <m:r>
      <m:t>1</m:t>
    </m:r>
  </m:sub>
  <m:sup>
    <m:r>
      <m:t>2</m:t>
    </m:r>
  </m:sup>
  <m:e>
    <m:r>
      <m:t>A</m:t>
    </m:r>
  </m:e>
</m:sPre>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.113 subHide (Hide Subscript (n-ary))

This element specifies the n-ary Hide Subscript property. When 1 or `true`, the lower limit does not appear, as in $\int^{\infty} \frac{x}{x+1} dx$. If this element is omitted, the lower limit appears.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

When this element is omitted, the default is `false`. In other words, When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: An example of this element in use is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:subHide m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.114 sup (Superscript (Superscript object))

This element specifies the superscript of the superscript object sSup. For example, the sup in the superscript object x^n is n. [*Example*: An example of this element in use is:

```
<m:sSup>
  <m:e>
    <m:r>
      <m:t>x</m:t>
    </m:r>
  </m:e>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <m:sup>
      <m:r>
        <m:t>n</m:t>
      </m:r>
    </m:sup>
  </m:sSup>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_OMathArg) is located in §A.6.1. *end note*]

22.1.2.115 supHide (Hide Superscript (n-ary))

This element specifies the n-ary Hide Superscript property. When `1` or `true`, the upper limit does not appear, as in $\int_0 \frac{x}{x+1} dx$. If this element is omitted, the lower limit appears.

When this element is omitted, the default is `false`. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: An example of this element in use is:

```
<m:naryPr>
  <m:chr  m:val="&#8747;"/>
  <m:supHide m:val="1"/>
</m:naryPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.1.2.116 t (Text)

This element specifies the text in a math run r.

[*Example*:

$J + K + a + j$ is represented by the following XML:

```
<m:oMath>
  <m:e>
    <m:r>
      <m:t>J+K+a+j</m:t>
    </m:r>
  </m:e>
</m:oMath>
```

end example]

Attributes	Description

xml:space (Content Contains Significant Whitespace)
 	
Namespace: http://www.w3.org/XML/1998/namespace
	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. [*Example*: Consider the following run contained within a WordprocessingML document:
	
	`<w:r>`
	`  <w:t>  significant whitespace  </w:t>`
	`</w:r>`
	
	Although there are three spaces on each side of the text content in the run, that whitespace has not been specifically marked as significant, therefore it is subject to the space preservation rules currently specified in that run's scope. *end example*]
	
	The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Text) is located in §A.6.1. *end note*]

22.1.2.117 transp (Transparent (Phantom))

This element specifies that the phantom is transparent for spacing. This means that if the contents of the phantom are belonging to a special spacing class (such as binary operators, relational operators, differentials, etc.), the contents of that phantom are taken into consideration when laying out text. If transparency is turned off, then the contents of the phantom are ignored during layout. When this element is omitted, transparency is `0` or `false`. In other words, when the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

In the following example, transparency is not applied on the phantom around the differential term. $\int xdx$. The spacing is incorrect. In the following integral, the only difference is that transparency is applied: $\int x\,dx$. Now the spacing is correct.

[*Example*: An example of this element in XML is:

```
<m:phantPr>
  <m:zeroAsc  m:val="1"/>
  <m:zeroDesc  m:val="1"/>
  <m:transp  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.118 type (Fraction type)

This element specifies the type of fraction f; the default is 'bar'. Whether the element is absent or present without the val attribute, the default of the val attribute is bar. Fractions types are:

Stacked Fraction: $\frac{a}{b}$

Skewed Fraction: a/b

Linear Fraction: a/b

Stack Object (No-Bar Fraction): $\binom{n}{k}$

[Example:

c/d is represented by the following XML:

```
<m:f>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <m:fPr>
      <m:type m:val="skw"/>
    </m:fPr>
    <m:num>
      <m:r>
        <m:t>c</m:t>
      </m:r>
    </m:num>
    <m:den>
      <m:r>
        <m:t>d</m:t>
      </m:r>
    </m:den>
  </m:f>
```

end example]

Attributes	Description
val (Value)	Specifies the type of fraction. Possible values are `bar` (Bar Fraction), `lin` (Linear Fraction), `noBar` (No-Bar Fraction (Stack)), and `skw` (Skewed). The possible values for this attribute are defined by the ST_FType simple type (§22.1.3.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_FType) is located in §A.6.1. *end note*]

22.1.2.119 vertJc (Vertical Justification)

This element, combined with pos of groupChrPr, specifies the vertical layout of the groupChr object. Where pos specifies the position of the grouping character, vertJc specifies the alignment of the object with respect to the baseline. For example, when the group character is above the object, vertJc of `top` signifies that the top of the object falls on the baseline; when vertJc is set to `bot`, the bottom of the object is on the baseline. The table below demonstrates the four possible combinations of groupChr layout:

Pos	vertJc	Layout
`top`	`top`	$a \underset{bcd}{\longleftarrow} e$
`top`	`bot`	$a \overbrace{bcd} e$
`bot`	`top`	$a \underbrace{bcd} e$
`bot`	`bot`	$a \overset{yields}{\longrightarrow} b$

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

When the element is absent, the default value of the property is top.

When the element is present and the val attribute is absent, the default of the val attribute is bot.

[*Example*:

$a + \underbrace{b+c} + d$ is represented by the following XML:

```
<m:oMath>
  <m:r>
    <m:t>a+</m:t>
  </m:r>
  <m:groupChr>
    <m:groupChrPr>
      <m:vertJc m:val="bot"/>
    </m:groupChrPr>
    <m:e>
      <m:r>
        <m:t>b+c</m:t>
      </m:r>
    </m:e>
  </m:groupChr>
  <m:r>
    <m:t>+d</m:t>
  </m:r>
</m:oMath>
```

Since the vertical justification property is set to bot on the group-character object, the bottom of the group-character object is aligned with the baseline.

end example]

Attributes	Description
val (Value)	Specifies the position of the parent element. Possible values are top and bot. [*Example*: <m:barPr> <m:pos m:val="top"/> </m:barPr> The possible values for this attribute are defined by the ST_TopBot simple type (§22.1.3.13).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_TopBot) is located in §A.6.1. *end note*]

22.1.2.120 wrapIndent (Wrap Indent)

This element specifies the indent of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. The default indent is 1". In other words, whether the element is absent or present without the val attribute, the default of the val attribute is 1440 twips (or 1 inch).

[*Example*: The XML below demonstrates wrapIndent in use:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapIndent m:val="1440"/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (Value)	Specifies the value, in twips, of the parent element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§22.9.2.14).

[*Note*: The W3C XML Schema definition of this element's content model (CT_TwipsMeasure) is located in §A.6.1. *end note*]

22.1.2.121 wrapRight (Wrap Right)

This element specifies the right justification of the wrapped line of an instance of mathematical text. The line or lines of a wrapped instance of mathematical text after the line break can either be indented by a specified amount from the left margin, or right aligned. If this element is present, the continuation is right aligned.

When the element is absent, the default value of the option is 0 meaning that this option is not applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this option is applied.

[*Example*: An example of this element in use is:

```
<m:mathPr>
  <m:mathFont m:val="Cambria Math"/>
  <m:brkBin m:val="before"/>
  <m:brkBinSub m:val="--"/>
  <m:smallFrac m:val="0"/>
  <m:dispDef/>
  <m:lMargin m:val="0"/>
  <m:rMargin m:val="0"/>
  <m:defJc m:val="centerGroup"/>
  <m:wrapRight/>
  <m:intLim m:val="subSup"/>
  <m:naryLim m:val="undOvr"/>
</m:mathPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.122 zeroAsc (Phantom Zero Ascent)

This element specifies that the phantom has zero ascent. The ascent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have ascent (zero ascent is not applied).

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

In the following example, the differential term is contained in a phantom that zero ascent. As a result, spacing is reduced between the tip of the "d" and the radical bar: $\sqrt{xdx}$. [*Example*:

```
<m:phantPr>
  <m:zeroAsc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of `1` or `true` specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of `0` or `false` specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.123 zeroDesc (Phantom Zero Descent)

This element specifies that the phantom has zero descent. The descent of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have descent (zero descent is not applied).

When the element is absent, the default value of the property is `0` meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is `1` meaning that this property is applied.

[*Example*: In the following two examples, only the second has zero descent around the "y". Note that the radical is smaller than in the first case. $\sqrt{y}\sqrt{y}$

```
<m:phantPr>
  <m:zeroDesc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.2.124 zeroWid (Phantom Zero Width)

This element specifies that the phantom has zero width. The width of the contents of the phantom is not taken into account during layout. When this property is omitted, the phantom does have width (zero width is not applied).

When the element is absent, the default value of the property is 0 meaning that this property is not applied.

When the element is present and the val attribute is absent, the default of the val attribute is 1 meaning that this property is applied.

[*Example*: In the folowing example, the second radical contains a phantom of the fraction a/b. The phantom has zero width, such that only the height grows to accommodate the hidden fraction: $\sqrt{\frac{a}{b}} = \sqrt{x}$

```
<m:phantPr>
  <m:show  m:val="0"/>
  <m:zeroDesc  m:val="1"/>
</m:phantPr>
```

end example]

Attributes	Description
val (value)	Specifies a binary value for the property defined by the parent XML element. A value of 1 or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present. A value of 0 or false specifies that the property shall be explicitly turned off. This is implied when the parent element is not present. The possible values for this attribute are defined by the ST_OnOff simple type (§22.9.2.7).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_OnOff) is located in §A.6.1. *end note*]

22.1.3 Simple Types

This is the complete list of simple types dedicated to Math.

22.1.3.1 ST_BreakBin (Break Binary Operators)

This defines how to represent binary operators with respect to a line-wrapping break. The line can wrap before the operator or after the operator; alternately, the operator can appear both at the end of the first line and the beginning of the second.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
after (After)	When line-wrapping breaks occur on binary operators, the binary operator appears after the break (at the start of the next line).
before (Before)	When line-wrapping breaks occur on binary operators, the binary operator appears before the break (at the end of the first line).
repeat (Repeat)	When line-wrapping breaks occur on binary operators, the binary operator appears on both sides of the break (at the end of the first line and the start of the next line).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BreakBin) is located in §A.6.1. *end note*]

22.1.3.2 ST_BreakBinSub (Break on Binary Subtraction)

This simple type specifies how to represent subtraction on both sides of a line-wrapping break, when the Break Binary Operators option is set to repeat. The first character represents the sign at the end of the line with the break; the second represents the sign at the start of the wrapped line. Options are --, -+, and +-.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
+- (Plus Minus)	Repetition of subtraction sign after a line-wrapping break is plus on the first line and minus on the second line.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
-+ (Minus Plus)	Repetition of subtraction sign after a line-wrapping break is minus on the first line and plus on the second line.
-- (Minus Minus)	Repetition of subtraction sign after a line-wrapping break is minus on the first and second lines.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_BreakBinSub) is located in §A.6.1. *end note*]

22.1.3.3 ST_Char (Character)

This Simple Type specifies the single character used by the parent element.

[*Example*: In the following example, $\{a\}$ uses { and } as its enclosing characters, instead of the default (and).

```
<m:dPr>
  <m:begChr  m:val="{"/>
  <m:endChr m:val="}"/>
</m:dPr>
```

end example]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 1 characters.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Char) is located in §A.6.1. *end note*]

22.1.3.4 ST_FType (Fraction Type)

Fractions can be of type bar (horizontal fraction bar), skewed ("skw" - diagonal fraction bar with kerned and vertically adjusted numerator and denominator), linear ("lin" - diagonal fraction bar, takes up exactly one line of space), and the "stack" object ("noBar").

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bar (Bar Fraction)	Fraction with a horizontal fraction bar.
lin (Linear Fraction)	Fraction with slanted fraction bar, that takes up no

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	additional vertical space.
noBar (No-Bar Fraction (Stack))	Stack object, which looks like a fraction with no fraction bar.
skw (Skewed)	Fraction with diagonal fraction bar.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FType) is located in §A.6.1. *end note*]

22.1.3.5 ST_Integer2 (Integer value (-2 to 2))

This simple type contains a value from (-2,+2) which specifies the size of the argument. The effects of each value are described by the referencing element.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to -2.
- This simple type has a maximum value of less than or equal to 2.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Integer2) is located in §A.6.1. *end note*]

22.1.3.6 ST_Integer255 (Integer value (1 to 255))

This simple type specifies an integer value. The semantics of each value are discussed by the referencing element.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 1.
- This simple type has a maximum value of less than or equal to 255.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Integer255) is located in §A.6.1. *end note*]

22.1.3.7 ST_Jc (Justification)

This Simple Type specifies the justification of Math Paragraphs. Justification of the Math Paragraph can be Left, Right, Centered, or Centered as Group.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Center (Text))	Centers each instance of mathematical text individually with respect to margins.
centerGroup (Centered as Group (Text))	Justifies instances of mathematical text with respect to each other, and centers the group of mathematical text (the Math Paragraph) with respect to the page.
left (Left Justification)	Left justification of Math Paragraph
right (Right)	Right Justification of Math Paragraph

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Jc) is located in §A.6.1. *end note*]

22.1.3.8 ST_LimLoc (Limit Location)

Limits can be in one of two positions: Under-Over (`undOvr` - above and below the base), and Subscript-Superscript (`subSup` - positioned to the side of the base, in the position of subscripts and superscripts).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
subSup (Subscript-Superscript location)	Limits placed to the side of the base, as opposed to directly over and under.
undOvr (Under-Over location)	Limits placed to the directly above and below the base, as opposed to on the side.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_LimLoc) is located in §A.6.1. *end note*]

22.1.3.9 ST_Script (Script)

Script can be of type Roman, Script, Fraktur, Double-Struck, Sans-Serif, or Monospace.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
double-struck (double-struck)	Double-Struck Script Type
fraktur (Fraktur)	Fraktur Script Type
monospace (Monospace)	Monospace Script Type

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
roman (Roman)	Roman Script Type
sans-serif (Sans-Serif)	Sans-Serif Script Type
script (Script)	Script Type

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Script) is located in §A.6.1. *end note*]

22.1.3.10 ST_Shp (Shape (Delimiters))

Delimiters shape can be centered around the argument, or matched to the shape of the argument.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
centered (Centered (Delimiters))	Delimiters are centered around their argument.
match (Match)	Match shape of contents of delimiters.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Shp) is located in §A.6.1. *end note*]

22.1.3.11 ST_SpacingRule (Spacing Rule)

Integer value (0 to 4), representing the type of spacing between rows.

This simple type's contents are a restriction of the W3C XML Schema integer datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 4.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SpacingRule) is located in §A.6.1. *end note*]

22.1.3.12 ST_Style (Style)

Style of math can be plain, bold, italic, or bold-italic (p, bi, i, or bi).

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
b (Bold)	Bold
bi (Bold-Italic)	Bold-Italic
i (Italic)	Italic
p (Plain)	Plain

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Style) is located in §A.6.1. *end note*]

22.1.3.13 ST_TopBot (Top-Bottom)

Possible values are `top` and `bot`.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bot (Bottom Alignment)	Aligns the bottom of the object to the baseline of the surrounding text.
top (Top)	Aligns the top row of the object to the baseline of the surrounding text.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TopBot) is located in §A.6.1. *end note*]

22.1.3.14 ST_UnSignedInteger (Unsigned integer.)

Unsigned Integer

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnSignedInteger) is located in §A.6.1. *end note*]

22.2 Extended Properties

Extended properties are a predefined set of metadata properties that are applicable to Office Open XML documents. These properties extend the set of core properties defined in Part 2: "Open Packaging Conventions" which are common to all packages.

Extended properties are stored within an Extended File Properties part with:

- Source Relationship: http://purl.oclc.org/ooxml/officeDocument/relationships/extendedProperties

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

- Content Type: application/vnd.openxmlformats-officedocument.extended-properties+xml

Each extended property is represented as an element in the extended properties part. Extended properties elements are non-repeatable and can be empty or omitted. If all extended property elements are omitted then the extended properties part can be excluded from a document.

[*Example*: A sample extended file properties part:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Properties xmlns="http://…/extended-properties"
  xmlns:vt="http://…/docPropsVTypes">
..<Template>Sales Report.dotm</Template>
..<TotalTime>0</TotalTime>
..<Pages>1</Pages>
..<Words>166</Words>
..<Characters>948</Characters>
..<Application>Microsoft Office Word</Application>
..<DocSecurity>0</DocSecurity>
..<Lines>7</Lines>
..<Paragraphs>2</Paragraphs>
..<ScaleCrop>false</ScaleCrop>
..<Company>Northwind Traders</Company>
..<LinksUpToDate>false</LinksUpToDate>
..<CharactersWithSpaces>1112</CharactersWithSpaces>
..<SharedDoc>false</SharedDoc>
..<HyperlinksChanged>false</HyperlinksChanged>
..<AppVersion>12.0000</AppVersion>
</Properties>
```

end example]

22.2.1 Table of Contents

This subclause is informative.

22.2.2 Elements 3722
22.2.2.1 Application (Application Name) 3722
22.2.2.2 AppVersion (Application Version) 3722
22.2.2.3 Characters (Total Number of Characters) 3722
22.2.2.4 CharactersWithSpaces (Number of Characters (With Spaces)) 3723
22.2.2.5 Company (Name of Company) 3723
22.2.2.6 DigSig (Digital Signature) 3723
22.2.2.7 DocSecurity (Document Security) 3723
22.2.2.8 HeadingPairs (Heading Pairs) 3723
22.2.2.9 HiddenSlides (Number of Hidden Slides) 3724
22.2.2.10 HLinks (Hyperlink List) 3724

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.2.2.11 HyperlinkBase (Relative Hyperlink Base) 3724
22.2.2.12 HyperlinksChanged (Hyperlinks Changed) 3724
22.2.2.13 Lines (Number of Lines) 3725
22.2.2.14 LinksUpToDate (Links Up-to-Date) 3725
22.2.2.15 Manager (Name of Manager) 3725
22.2.2.16 MMClips (Total Number of Multimedia Clips) 3725
22.2.2.17 Notes (Number of Slides Containing Notes) 3725
22.2.2.18 Pages (Total Number of Pages) 3725
22.2.2.19 Paragraphs (Total Number of Paragraphs) 3725
22.2.2.20 PresentationFormat (Intended Format of Presentation) 3725
22.2.2.21 Properties (Application Specific File Properties) 3726
22.2.2.22 ScaleCrop (Thumbnail Display Mode) 3726
22.2.2.23 SharedDoc (Shared Document) 3726
22.2.2.24 Slides (Slides Metadata Element) 3726
22.2.2.25 Template (Name of Document Template) 3726
22.2.2.26 TitlesOfParts (Part Titles) 3726
22.2.2.27 TotalTime (Total Edit Time Metadata Element) 3727
22.2.2.28 Words (Word Count) 3727

End of informative text.

22.2.2 Elements

The following elements specify the contents of this namespace:

22.2.2.1 Application (Application Name)

This element specifies the name of the application that created this document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.2 AppVersion (Application Version)

This element specifies the version of the application which produced this document.

The content of this element shall be of the form XX.YYYY where X and Y represent numerical values, or the document shall be considered non-conformant.

[*Note*: The contents of this element do not represent absolute values, but rather qualify the contents of the Application element to differentiate between different versions of the same producer. Applications should use this information in an informative manner only (as document metadata). *end note*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.3 Characters (Total Number of Characters)

This element specifies the total number of characters in a document.

The possible values for this element are defined by the W3C XML Schema int datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.2.2.4 CharactersWithSpaces (Number of Characters (With Spaces))

This element specifies the last count of the number of characters (including spaces) in this document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.5 Company (Name of Company)

This element specifies the name of a company associated with the document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.6 DigSig (Digital Signature)

This element contains the signature of a digitally signed document.

[*Note*: This property is a mechanism used by legacy documents to store the digital signature of its binary representation, and should be avoided in favor of the well-defined mechanism defined in Part 2. Any use of this property should be for legacy compatibility only, and is application-defined. *end note*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DigSigBlob) is located in §A.6.2. *end note*]

22.2.2.7 DocSecurity (Document Security)

This metadata element specifies the security level of a document as a numeric value. Document security is defined as:

DocSecurity	Security Level
1	Document is password protected.
2	Document is recommended to be opened as read-only.
4	Document is enforced to be opened as read-only.
8	Document is locked for annotation.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.8 HeadingPairs (Heading Pairs)

Heading pairs indicates the grouping of document parts and the number of parts in each group. These parts are not document parts but conceptual representations of document sections.

[*Example*: A presentation composing of three slides with an applied theme can have the following HeadingPairs representation:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
<HeadingPairs>
  <vt:vector size="4" baseType="variant">
    <vt:variant>
      <vt:lpstr>Theme</vt:lpstr>
    </vt:variant>
    <vt:variant>
      <vt:i4>1</vt:i4>
    </vt:variant>
    <vt:variant>
      <vt:lpstr>Slide Titles</vt:lpstr>
    </vt:variant>
    <vt:variant>
      <vt:i4>3</vt:i4>
    </vt:variant>
  </vt:vector>
</HeadingPairs>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorVariant) is located in §A.6.2. *end note*]

22.2.2.9 HiddenSlides (Number of Hidden Slides)

This element specifies the number of hidden slides in a presentation document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.10 HLinks (Hyperlink List)

This element specifies the set of hyperlinks that were in this document when last saved.

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorVariant) is located in §A.6.2. *end note*]

22.2.2.11 HyperlinkBase (Relative Hyperlink Base)

This element specifies the base string used for evaluating relative hyperlinks in this document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.12 HyperlinksChanged (Hyperlinks Changed)

This element specifies that one or more hyperlinks in this part were updated exclusively in this part by a producer. The next producer to open this document shall update the hyperlink relationships with the new hyperlinks specified in this part.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.2.2.13 Lines (Number of Lines)

This element specifies the total number of lines in a document when last saved by a conforming producer if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.14 LinksUpToDate (Links Up-to-Date)

This element indicates whether hyperlinks in a document are up-to-date. Set this element to TRUE to indicate that hyperlinks are updated. Set this element to FALSE to indicate that hyperlinks are outdated.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.15 Manager (Name of Manager)

This element specifies the name of a supervisor associated with the document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.16 MMClips (Total Number of Multimedia Clips)

This element specifies the total number of sound or video clips that are present in the document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.17 Notes (Number of Slides Containing Notes)

This element specifies the number of slides in a presentation containing notes.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.18 Pages (Total Number of Pages)

This element specifies the total number of pages of a document if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.19 Paragraphs (Total Number of Paragraphs)

This element specifies the total number of paragraphs found in a document if applicable.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.20 PresentationFormat (Intended Format of Presentation)

This element specifies the intended format for a presentation document. For example, a presentation intended to be shown on video has PresentationFormat "Video".

The possible values for this element are defined by the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.2.2.21 Properties (Application Specific File Properties)

This element specifies the application properties of a document. For properties of type string, NCR escape format (_xHHHH_) is used for any invalid XML characters.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Properties) is located in §A.6.2. *end note*]

22.2.2.22 ScaleCrop (Thumbnail Display Mode)

This element indicates the display mode of the document thumbnail. Set this element to TRUE to enable scaling of the document thumbnail to the display. Set this element to FALSE to enable cropping of the document thumbnail to show only sections that fits the display.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.23 SharedDoc (Shared Document)

This element indicates if this document is currently shared between multiple producers. If this element is set to TRUE, producers should take care when updating the document.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

22.2.2.24 Slides (Slides Metadata Element)

This element specifies the total number of slides in a presentation document.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.25 Template (Name of Document Template)

This element specifies the name of an external document template containing format and style information used to create the current document.

The possible values for this element are defined by the W3C XML Schema string datatype.

22.2.2.26 TitlesOfParts (Part Titles)

This element specifies the title of each document. These parts are not document parts but conceptual representations of document sections.

[*Example*: A presentation composing of three slides with an applied theme "Currency" can have the following TitlesofParts representation:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<TitlesofParts>
  <vt:vector size="4" baseType="lpstr">
    <vt:lpstr>Currency</vt:lpstr>
    <vt:lpstr>Slide 1</vt:lpstr>
    <vt:lpstr>Slide 2</vt:lpstr>
    <vt:lpstr>Slide 3</vt:lpstr>
  </vt:vector>
</TitlesofParts>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_VectorLpstr) is located in §A.6.2. *end note*]

22.2.2.27 TotalTime (Total Edit Time Metadata Element)

Total time that a document has been edited. The default time unit is minutes.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.2.2.28 Words (Word Count)

This element specifies the total number of words contained in a document when last saved.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.3 Custom Properties

Custom properties enable users to define custom metadata properties through a set of well-defined data types.

Custom properties are represented by property elements (§22.3.2.2) stored in the Custom File Properties part with:

- Source Relationship: http://purl.oclc.org/ooxml/officeDocument/relationships/customProperties
- Content Type: application/vnd.openxmlformats-officedocument.custom-properties+xml

Custom property elements are non-repeatable and can be empty or omitted. If all custom property elements are omitted then the custom properties part can be excluded from a document.

The type and value of custom properties are specified by child XML elements in the File Properties Variant Types namespace (discussed in detail in §22.4). User defined properties are uniquely identified through the name attribute of the property element. Custom properties can be associated with OLE document properties through the fmtid and pid attributes.

[*Example*: A custom OLE `Editor` property of type `string` can be defined as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<property fmtid="{D5CDD505-2E9C-101B-9397-08002B2CF9AE}" pid="2"
  name="Editor">
  <vt:lpwstr>John Smith</vt:lpwstr>
</property>
```

end example]

22.3.1 Table of Contents

This subclause is informative.

22.3.2 Elements 3728
22.3.2.1 Properties (Custom File Properties) 3728
22.3.2.2 property (Custom File Property) 3728

End of informative text.

22.3.2 Elements

This subclause specifies the set of elements that define this namespace:

22.3.2.1 Properties (Custom File Properties)

Parent element for the custom file properties part.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Properties) is located in §A.6.3. *end note*]

22.3.2.2 property (Custom File Property)

This element specifies a single custom file property. Custom file property type is defined through child elements in the File Properties Variant Type namespace. Custom file property value can be set by setting the appropriate Variant Type child element value.

Attributes	Description
fmtid (Format ID)	Uniquely relates a custom property with an OLE property. The value of this attribute is a Globally Unique Identifier in the form of {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHH} where each H is a hexidecimal. The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).
linkTarget (Bookmark Link Target)	Specifies the name of a bookmark in the current document (for WordprocessingML), or a table or named cell (for SpreadsheetML) from which the value of this custom document property should be extracted. If this attribute is present, then any value under this element shall be considered a cache and replaced with the value of this bookmark (if present) on save. If the bookmark is not present, then this link shall be considered broken and the cached value shall be retained.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.
name (Custom File Property Name)	Specifies the name of this custom file property. The possible values for this attribute are defined by the W3C XML Schema string datatype.
pid (Property ID)	Uniquely relates a custom property with an OLE property. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Property) is located in §A.6.3. *end note*]

22.4 Variant Types

This subclause specifies the set of data types which can be included within file properties that accept variant type structures.

22.4.1 Table of Contents

This subclause is informative.

22.4.2 Elements 3730
22.4.2.1 array (Array) 3730
22.4.2.2 blob (Binary Blob) 3731
22.4.2.3 bool (Boolean) 3731
22.4.2.4 bstr (Basic String) 3732
22.4.2.5 clsid (Class ID) 3732
22.4.2.6 cy (Currency) 3732
22.4.2.7 date (Date and Time) 3732
22.4.2.8 decimal (Decimal) 3732
22.4.2.9 empty (Empty) 3732
22.4.2.10 error (Error Status Code) 3733
22.4.2.11 filetime (File Time) 3733
22.4.2.12 i1 (1-Byte Signed Integer) 3733
22.4.2.13 i2 (2-Byte Signed Integer) 3733
22.4.2.14 i4 (4-Byte Signed Integer) 3733
22.4.2.15 i8 (8-Byte Signed Integer) 3733
22.4.2.16 int (Integer) 3733
22.4.2.17 lpstr (LPSTR) 3733
22.4.2.18 lpwstr (LPWSTR) 3734
22.4.2.19 null (Null) 3734
22.4.2.20 oblob (Binary Blob Object) 3734
22.4.2.21 ostorage (Binary Storage Object) 3734

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.4.2.22 ostream (Binary Stream Object) 3734
22.4.2.23 r4 (4-Byte Real Number) 3735
22.4.2.24 r8 (8-Byte Real Number) 3735
22.4.2.25 storage (Binary Storage) 3735
22.4.2.26 stream (Binary Stream) 3735
22.4.2.27 ui1 (1-Byte Unsigned Integer) 3735
22.4.2.28 ui2 (2-Byte Unsigned Integer) 3735
22.4.2.29 ui4 (4-Byte Unsigned Integer) 3735
22.4.2.30 ui8 (8-Byte Unsigned Integer) 3735
22.4.2.31 uint (Unsigned Integer) 3736
22.4.2.32 variant (Variant) 3736
22.4.2.33 vector (Vector) 3736
22.4.2.34 vstream (Binary Versioned Stream) 3737

22.4.3 Simple Types 3737

22.4.3.1 ST_ArrayBaseType (Array Base Type Simple Type) 3737
22.4.3.2 ST_Cy (Currency Simple Type) 3738
22.4.3.3 ST_Error (Error Status Code Simple Type) 3739
22.4.3.4 ST_VectorBaseType (Vector Base Type Simple Type) 3739

End of informative text.

22.4.2 Elements

The following elements define the contents of this namespace:

22.4.2.1 array (Array)

The array element defines the array variant type. Array contents shall be of uniform type as specified by the baseType attribute. The contents of an array are defined using repeated child elements of the appropriate variant type.

Multi-dimensional arrays can be defined by specifying the length of each dimension in the lBound and uBound attributes through the use of the "," delimiter. Child elements of multi-dimensional arrays are indexed along each dimension in the order the dimensions are declared.

In other words, the array shall be filled as follows:

- The first index shall be incremented to its maximum value [*Example*: [0,0,0] to [max,0,0] *end example*]
- The second index shall be incremented to its maximum value [*Example*: [0,1,0] to [0,max,0] *end example*]
- Subsequent indices shall be filled until all provided values have been added
- All other values shall have null values within the array (i.e. no default value shall be assumed).

[*Example*: A 2x3 variant type array of type "i4" is specified as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<vt:array lBounds="0,0" uBounds="1,2" baseType="i4">
  <vt:i4>0</vt:i4>
  <vt:i4>1</vt:i4>
  <vt:i4>2</vt:i4>
  <vt:i4>3</vt:i4>
  <vt:i4>4</vt:i4>
</vt:array>
```

The resulting array: [0,0] = 0, [1,0] = 1, [0,1] = 2, [1,1] = 3, [0,2] = 4. *end example*]

Attributes	Description
baseType (Array Base Type)	The baseType attribute specifies the base variant type of an array. The allowed values are: variant, i1, i2, i4, int, ui1, ui2, ui4, uint, r4, r8, decimal, bstr, date, bool, cy, and error. The possible values for this attribute are defined by the ST_ArrayBaseType simple type (§22.4.3.1).
lBounds (Array Lower Bounds Attribute)	The lBounds attribute specifies the lower bound of an array in the format: #, #, # … # where each # represents an integer. The possible values for this attribute are defined by the W3C XML Schema int datatype.
uBounds (Array Upper Bounds Attribute)	The uBounds attribute specifies the upper bound of an array in the format: #, #, # … # where each # represents an integer. The possible values for this attribute are defined by the W3C XML Schema int datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Array) is located in §A.6.4. *end note*]

22.4.2.2 blob (Binary Blob)

This element specifies a base64 binary blob variant type.

This type is defined as follows: a DWORD count of bytes, followed by that many bytes of data. The byte count does not include the four bytes for the length of the count itself; an empty blob member would have a count of zero, followed by zero bytes.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.3 bool (Boolean)

This element specifies a Boolean variant type.

The possible values for this element are defined by the W3C XML Schema boolean datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.4.2.4 bstr (Basic String)

This element defines a binary basic string variant type, which can store any valid Unicode character. Unicode characters that cannot be directly represented in XML as defined by the XML 1.0 specification, shall be escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it shall be escaped as _x0008_. *end example*] To store the literal form of an escape sequence, the initial underscore shall itself be escaped (i.e. stored as _x005F_). [*Example*: The string literal `_x0008_` would be stored as `_x005F_x0008_`. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.5 clsid (Class ID)

This element specifies a class ID variant type. The value shall be a Globally Unique Identifier with format: {HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHH}.

The possible values for this element are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Guid) is located in §A.6.9. *end note*]

22.4.2.6 cy (Currency)

This element specifies a currency variant type with exactly four digits after the decimal point.

The possible values for this element are defined by the ST_Cy simple type (§22.4.3.2).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Cy) is located in §A.6.4. *end note*]

22.4.2.7 date (Date and Time)

This element specifies a date variant type of type date-time as defined in RFC 3339.

The possible values for this element are defined by the W3C XML Schema dateTime datatype.

22.4.2.8 decimal (Decimal)

This element specifies a decimal variant type.

The possible values for this element are defined by the W3C XML Schema decimal datatype.

22.4.2.9 empty (Empty)

This element specifies an empty variant type. No values or child elements are allowed.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Empty) is located in §A.6.4. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.4.2.10 error (Error Status Code)

The error element specifies a 32-bit error status code variant type of the form 0xHHHHHHHH. Each H represents a hexadecimal digit.

The possible values for this element are defined by the ST_Error simple type (§22.4.3.3).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Error) is located in §A.6.4. *end note*]

22.4.2.11 filetime (File Time)

This element specifies a file-time variant type of type date-time as defined in RFC 3339.

The possible values for this element are defined by the W3C XML Schema dateTime datatype.

22.4.2.12 i1 (1-Byte Signed Integer)

This element specifies a 1-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema byte datatype.

22.4.2.13 i2 (2-Byte Signed Integer)

This element specifies a 2-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema short datatype.

22.4.2.14 i4 (4-Byte Signed Integer)

This element specifies a 4-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.4.2.15 i8 (8-Byte Signed Integer)

This element specifies a 8-byte signed integer variant type.

The possible values for this element are defined by the W3C XML Schema long datatype.

22.4.2.16 int (Integer)

This element specifies an integer variant type.

The possible values for this element are defined by the W3C XML Schema int datatype.

22.4.2.17 lpstr (LPSTR)

This element specifies a string variant type. For all characters that cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.18 lpwstr (LPWSTR)

This element specifies a string variant type. For all characters that cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *end example*]

The possible values for this element are defined by the W3C XML Schema string datatype.

22.4.2.19 null (Null)

This element specifies a null variant type.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Null) is located in §A.6.4. *end note*]

22.4.2.20 oblob (Binary Blob Object)

This element specifies a base64 binary blob object variant type.

This type is defined as follows: A blob member that contains a serialized object in the same representation that would appear in the ostream element. That is, a DWORD byte count (where the byte count does not include the size of itself) which is in the format of a class identifier followed by initialization data for that class.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.21 ostorage (Binary Storage Object)

This element specifies a base64 binary storage object variant type.

This type is defined as follows: Identical to the storage element, but indicates that the designated storage shall contain a loadable object.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.22 ostream (Binary Stream Object)

This element specifies a binary stream object variant type.

This type is defined as follows: Identical to the definition of the stream element, but indicates that the stream contains a serialized object, which is a CLSID – see the ST_Guid simple type (§22.9.2.4) – followed by initialization data for the specified class.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.4.2.23 r4 (4-Byte Real Number)

This element specifies a 4-byte real number variant type.

The possible values for this element are defined by the W3C XML Schema float datatype.

22.4.2.24 r8 (8-Byte Real Number)

This element specifies a 8-byte real number variant type.

The possible values for this element are defined by the W3C XML Schema double datatype.

22.4.2.25 storage (Binary Storage)

This element specifies a binary storage variant type.

This type is defined as follows: Contains the base64-encoded data for a structured storage.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.26 stream (Binary Stream)

This element specifies a binary stream variant type.

This type is defined as follows: Contains the base64-encoded data for a structured storage stream.

The possible values for this element are defined by the W3C XML Schema base64Binary datatype.

22.4.2.27 ui1 (1-Byte Unsigned Integer)

This element specifies a 1-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedByte datatype.

22.4.2.28 ui2 (2-Byte Unsigned Integer)

This element specifies a 2-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedShort datatype.

22.4.2.29 ui4 (4-Byte Unsigned Integer)

This element specifies a 4-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedInt datatype.

22.4.2.30 ui8 (8-Byte Unsigned Integer)

This element specifies a 8-byte unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedLong datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.4.2.31 uint (Unsigned Integer)

This element specifies an unsigned integer variant type.

The possible values for this element are defined by the W3C XML Schema unsignedInt datatype.

22.4.2.32 variant (Variant)

This element can contain exactly 1 child element of any variant type. This element is only valid as a child element of a vector or array variant type.

[*Example*: A vector of variant types:

```
<vt:vector baseType="variant">
  <vt:variant>
    <vt:i4>12</vt:i4>
  </vt:variant>
  <vt:variant>
    <vt:lpstr>WorkSheets</vt:lpstr>
  </vt:variant>
</vt:vector>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_Variant) is located in §A.6.4. *end note*]

22.4.2.33 vector (Vector)

This element defines the vector variant type. Vector contents shall be of uniform type as specified by the baseType attribute. The contents of a vector are defined using repeated child elements of the appropriate variant type.

[*Example*: A vector of lpstr variant types:

```
<vt:vector baseType="lpstr">
  <vt:lpstr>One</vt:lpstr>
  <vt:lpstr>Two</vt:lpstr>
  <vt:lpstr>Three</vt:lpstr>
</vt:vector>
```

end example]

Attributes	Description
baseType (Vector Base Type)	The baseType attribute specifies the base variant type of a vector. The allowed values are: variant, i1, i2, i4, i8, ui1, ui2, ui4, ui8, r4, r8, lpstr, lpwstr, bstr, date, filetime, bool, cy, error, and clsid.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the ST_VectorBaseType simple type (§22.4.3.4).
size (Vector Size)	Specifies the number of elements in the vector. The possible values for this attribute are defined by the W3C XML Schema unsignedInt datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vector) is located in §A.6.4. *end note*]

22.4.2.34 vstream (Binary Versioned Stream)

This element specifies a binary versioned stream variant type.

This type is defined as follows: A stream element's content with a GUID version (the version attribute).

[*Note*: The W3C XML Schema definition of this element's content model (CT_Vstream) is located in §A.6.4. *end note*]

22.4.3 Simple Types

This is the complete list of simple types dedicated to Variant Types.

22.4.3.1 ST_ArrayBaseType (Array Base Type Simple Type)

The ST_ArrayBaseType simple type defines the allowed values for an array's baseType attribute as: variant, i1, i2, i4, int, ui1,ui2, ui4, uint, r4, r8, decimal, bstr, date, bool, cy, and error.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bool (Boolean Base Type)	Specifies that the variant type for the contents of a array shall be `bool`.
bstr (Basic String Base Type)	Specifies that the variant type for the contents of a array shall be `bstr`.
cy (Curency Base Type)	Specifies that the variant type for the contents of a array shall be `cy`.
date (Date and Time Base Type)	Specifies that the variant type for the contents of a array shall be `date`.
decimal (Decimal Base Type)	Specifies that the variant type for the contents of a array shall be `decimal`.
error (Error Status Code Base Type)	Specifies that the variant type for the contents of a

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
	array shall be `error`.
i1 (1-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i1`.
i2 (2-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i2`.
i4 (4-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a array shall be `i4`.
int (Integer Base Type)	Specifies that the variant type for the contents of a array shall be `int`.
r4 (4-Byte Real Number Base Type)	Specifies that the variant type for the contents of a array shall be `r4`.
r8 (8-Byte Real Number Base Type)	Specifies that the variant type for the contents of a array shall be `r8`.
ui1 (1-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui1`.
ui2 (2-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui2`.
ui4 (4-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `ui4`.
uint (Unsigned Integer Base Type)	Specifies that the variant type for the contents of a array shall be `uint`.
variant (Variant Base Type)	Specifies that the variant type for the contents of a array shall be `variant`.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ArrayBaseType) is located in §A.6.4. *end note*]

22.4.3.2 ST_Cy (Currency Simple Type)

The ST_Cy simple type defines the cy element as a currency variant type with exactly four digits after the decimal point.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: \s*[0-9]*\.[0-9]{4}\s*.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Cy) is located in §A.6.4. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.4.3.3 ST_Error (Error Status Code Simple Type)

The ST_Error simple type defines a 32-bit error status code variant type of the form 0xHHHHHHHH. Each H represents a hexadecimal.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: \s*0x[0-9A-Za-z]{8}\s*.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Error) is located in §A.6.4. *end note*]

22.4.3.4 ST_VectorBaseType (Vector Base Type Simple Type)

The ST_VectorBaseType simple type defines the allowed values for a vector's baseType attribute as: variant, i1, i2, i4, i8, ui1, ui2, ui4, ui8, r4, r8, lpstr, lpwstr, bstr, date, filetime, bool, cy, error, and clsid.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bool (Boolean Base Type)	Specifies that the variant type for the contents of a vector shall be `bool`.
bstr (Basic String Base Type)	Specifies that the variant type for the contents of a vector shall be `bstr`.
clsid (Class ID Base Type)	Specifies that the variant type for the contents of a vector shall be `clsid`.
cy (Currency Base Type)	Specifies that the variant type for the contents of a vector shall be `cy`.
date (Date and Time Base Type)	Specifies that the variant type for the contents of a vector shall be `date`.
error (Error Status Code Base Type)	Specifies that the variant type for the contents of a vector shall be `error`.
filetime (File Time Base Type)	Specifies that the variant type for the contents of a vector shall be `filetime`.
i1 (Vector Base Type Enumeration Value)	Specifies that the variant type for the contents of a vector shall be `i1`.
i2 (2-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i2`.
i4 (4-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i4`.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
i8 (8-Byte Signed Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `i8`.
lpstr (LPSTR Base Type)	Specifies that the variant type for the contents of a vector shall be `lpstr`.
lpwstr (LPWSTR Base Type)	Specifies that the variant type for the contents of a vector shall be `lpwstr`.
r4 (4-Byte Real Number Base Type)	Specifies that the variant type for the contents of a vector shall be `r4`.
r8 (8-Byte Real Number Base Type)	Specifies that the variant type for the contents of a vector shall be `r8`.
ui1 (1-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui1`.
ui2 (2-Byte Unisigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui2`.
ui4 (4-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui4`.
ui8 (8-Byte Unsigned Integer Base Type)	Specifies that the variant type for the contents of a vector shall be `ui8`.
variant (Variant Base Type)	Specifies that the variant type for the contents of a vector shall be `variant`.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VectorBaseType) is located in §A.6.4. *end note*]

22.5 Custom XML Data Properties

This namespace defines the set of properties that can be associated with one or more custom XML parts within an Office Open XML document. A *custom XML part* is a part within an Office Open XML document, that contains arbitrary custom XML markup not necessarily defined by ISO/IEC 29500, and which is kept independent from the presentation-specific markup within the package.

[*Rationale*: It is often necessary to include custom XML semantics with an Office Open XML document, to store a complex set of properties (e.g., a document management system's metadata) along with the file. This mechanism allows this custom XML to be stored in the document in a way that is independent of the type of document and separate from the presentation markup—only the custom XML is stored in this part. *end rationale*]

The properties that can be applied to a custom XML part are:

- A part ID
- (optionally) One or more associated custom XML schemas

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.5.1 Table of Contents

This subclause is informative.

22.5.2 Elements 3741
22.5.2.1 datastoreItem (Custom XML Data Properties) 3741
22.5.2.2 schemaRef (Associated XML Schema) 3742
22.5.2.3 schemaRefs (Set of Associated XML Schemas) 3743

End of informative text.

22.5.2 Elements

The following information describes the elements in this namespace:

22.5.2.1 datastoreItem (Custom XML Data Properties)

This element specifies the properties for a single custom XML part inside of an Office Open XML document. The set of properties specified within this element are attached to the custom XML part that specifies a relationship to this part.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The datastoreItem element contains the properties for the custom XML part that referenced it; specifically, a part ID of A67AC88A-A164-4ADE-8889-8826CE44DE6E, and a single XML Schema reference to a schema with a target namespace of http://www.example.com/exampleSchema. *end example*]

Attributes	Description

itemID (Custom XML Data ID)	Specifies a globally unique identifier (GUID) that uniquely identifies a single custom XML part within an Office Open XML document.
	
	Each itemID value shall be unique among all custom XML data parts in this document. If a document contains duplicate itemID values, then the first value should be persisted, and subsequent values should be reassigned.
	
	[*Example*: Consider the following content for a custom XML part properties part:
	
	`<w:datastoreItem w:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">`
	`…`
	`</w:datastoreItem>`

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description
	The itemID attribute specifies that the ID associated with the parent custom XML part is A67AC88A-A164-4ADE-8889-8826CE44DE6E. *end example*] The possible values for this attribute are defined by the ST_Guid simple type (§22.9.2.4).

[*Note*: The W3C XML Schema definition of this element's content model (CT_DatastoreItem) is located in §A.6.5. *end note*]

22.5.2.2 schemaRef (Associated XML Schema)

This element specifies a single XML schema that is associated with the custom XML data part. This XML schema is identified using its target namespace, and can be located via any means available to an application processing the contents of this file.

If the custom XML part cannot be validated using the specified XML schema when it is opened, then this reference can be omitted when the file is subsequently saved.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The schemaRef element contains a single XML Schema reference to a schema with a target namespace of http://www.example.com/exampleSchema. Applications can then locate and utilize a schema for this namespace using any means available. *end example*]

Attributes	Description
uri (Target Namespace of Associated XML Schema)	Specifies the target namespace for the XML Schema associated with this schema reference. [*Example*: Consider the following content for a custom XML part properties part: … <w:schemaRef w:uri="http://www.example.com/schema1" /> <w:schemaRef w:uri="http://www.example.com/schema2" /> … The uri attribute specifies the target namespace of each XML schema reference: • http://www.example.com/schema1 • http://www.example.com/schema2

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Attributes	Description
	Applications can then locate and utilize a schema for these namespaces using any means available. *end example*] The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_DatastoreSchemaRef) is located in §A.6.5. *end note*]

22.5.2.3 schemaRefs (Set of Associated XML Schemas)

This element specifies the set of XML schemas that are associated with the parent custom XML part. Any number of XML schemas can be referenced, and this collection of schemas shall then be used to validate the contents of the corresponding custom XML part. If this element is present, then the set of XML schemas provided within should be used to validate the contents of the corresponding custom XML part (including the explicit presence of no child elements to specify that no custom XML schemas should be used even if one is present).

If this element is omitted, then applications can determine the set of XML schemas to be used to validate the contents of this part using any desired means.

[*Example*: Consider the following content for a custom XML part properties part:

```
<ds:datastoreItem ds:itemID="{A67AC88A-A164-4ADE-8889-8826CE44DE6E}">
  <ds:schemaRefs>
    <ds:schemaRef ds:uri="http://www.example.com/exampleSchema" />
  </ds:schemaRefs>
</ds:datastoreItem>
```

The schemaRefs element contains the set of XML Schema references that can be used to validate the contents of this part. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_DatastoreSchemaRefs) is located in §A.6.5. *end note*]

22.6 Bibliography

Within an Office Open XML document, it is possible to store an arbitrary amount of bibliographic data, the use of which can be determined by the application reading the content. This subclause defines the format and structure of that bibliographic data.

The outermost element of bibliographic data is Sources, which represents the collection of individual reference materials (Source) in the document.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.6.1 Table of Contents

This subclause is informative.

22.6.2 Elements 3745
22.6.2.1 AbbreviatedCaseNumber (Abbreviated Case Number) 3745
22.6.2.2 AlbumTitle (Album Title) 3746
22.6.2.3 Artist (Artist) 3746
22.6.2.4 Author (Contributors List) 3747
22.6.2.5 Author (Author) 3748
22.6.2.6 BookAuthor (Book Author) 3749
22.6.2.7 BookTitle (Book Title) 3749
22.6.2.8 Broadcaster (Broadcaster) 3750
22.6.2.9 BroadcastTitle (Broadcast Title) 3750
22.6.2.10 CaseNumber (Case Number) 3751
22.6.2.11 ChapterNumber (Chapter Number) 3751
22.6.2.12 City (City) 3751
22.6.2.13 Comments (Comments) 3752
22.6.2.14 Compiler (Compiler) 3752
22.6.2.15 Composer (Composer) 3753
22.6.2.16 Conductor (Conductor) 3753
22.6.2.17 ConferenceName (Conference or Proceedings Name) 3753
22.6.2.18 Corporate (Corporate Author) 3754
22.6.2.19 Counsel (Counsel) 3754
22.6.2.20 CountryRegion (Country or Region) 3754
22.6.2.21 Court (Court) 3755
22.6.2.22 Day (Day) 3755
22.6.2.23 DayAccessed (Day Accessed) 3756
22.6.2.24 Department (Department) 3756
22.6.2.25 Director (Director) 3756
22.6.2.26 Distributor (Distributor) 3757
22.6.2.27 Edition (Editor) 3757
22.6.2.28 Editor (Editor) 3757
22.6.2.29 First (Person's First, or Given, Name) 3758
22.6.2.30 Guid (GUID) 3758
22.6.2.31 Institution (Institution) 3759
22.6.2.32 InternetSiteTitle (Internet Site Title) 3759
22.6.2.33 Interviewee (Interviewee) 3760
22.6.2.34 Interviewer (Interviewer) 3760
22.6.2.35 Inventor (Inventor) 3761
22.6.2.36 Issue (Issue) 3761
22.6.2.37 JournalName (Journal Name) 3762
22.6.2.38 Last (Person's Last, or Family, Name) 3762
22.6.2.39 LCID (Locale ID) 3763
22.6.2.40 Medium (Medium) 3763
22.6.2.41 Middle (Person's Middle, or Other, Name) 3763
22.6.2.42 Month (Month) 3764
22.6.2.43 MonthAccessed (Month Accessed) 3764

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.44 NameList (Name List) 3764
22.6.2.45 NumberVolumes (Number of Volumes) 3765
22.6.2.46 Pages (Pages) 3765
22.6.2.47 PatentNumber (Patent Number) 3766
22.6.2.48 Performer (Performer) 3766
22.6.2.49 PeriodicalTitle (Periodical Title) 3766
22.6.2.50 Person (Person) 3767
22.6.2.51 ProducerName (Producer Name) 3767
22.6.2.52 ProductionCompany (Production Company) 3768
22.6.2.53 PublicationTitle (Publication Title) 3768
22.6.2.54 Publisher (Publisher) 3768
22.6.2.55 RecordingNumber (Recording Number) 3769
22.6.2.56 RefOrder (Reference Order) 3769
22.6.2.57 Reporter (Reporter) 3769
22.6.2.58 ShortTitle (Short Title) 3770
22.6.2.59 Source (Source) 3770
22.6.2.60 Sources (Sources) 3771
22.6.2.61 SourceType (Source Type) 3772
22.6.2.62 StandardNumber (Standard Number) 3773
22.6.2.63 StateProvince (State or Province) 3773
22.6.2.64 Station (Station) 3773
22.6.2.65 Tag (Tag) 3774
22.6.2.66 Theater (Theater) 3774
22.6.2.67 ThesisType (Thesis Type) 3775
22.6.2.68 Title (Title) 3775
22.6.2.69 Translator (Translator) 3775
22.6.2.70 Type (Patent Type) 3776
22.6.2.71 URL (URL) 3776
22.6.2.72 Version (Version) 3777
22.6.2.73 Volume (Volume) 3777
22.6.2.74 Writer (Writer) 3778
22.6.2.75 Year (Year) 3778
22.6.2.76 YearAccessed (Year Accessed) 3778
22.6.3 Simple Types 3779
22.6.3.1 ST_SourceType (Bibliographic Data Source Types) 3779

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

End of informative text.

22.6.2 Elements

The following elements define the contents of the Bibliography schema:

22.6.2.1 AbbreviatedCaseNumber (Abbreviated Case Number)

This element describes the abbreviated form of a case number. Typically, this field is used in the Case source type.

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.2 AlbumTitle (Album Title)

This element specifies the title of an album. Typically, this field is used in the Sound recording source type. [*Example*:

```
<b:Source>
  <b:Tag>Bon96</b:Tag>
  <b:SourceType>SoundRecording</b:SourceType>
  <b:Author>
    <b:Performer>
      <b:NameList>
        <b:Person>
          <b:Last>Villaron</b:Last>
          <b:First>Shawn</b:First>
          <b:Middle>Alan</b:Middle>
        </b:Person>
      </b:NameList>
    </b:Performer>
  </b:Author>
  <b:Title>Title</b:Title>
  <b:Year>2004</b:Year>
  <b:City>London</b:City>
  <b:AlbumTitle>Album Title</b:AlbumTitle>
  <b:RefOrder>15</b:RefOrder>
  <b:Guid>{17722923-790D-47E7-BB5D-C5DC67FA83D6}</b:Guid>
  <b:LCID>0</b:LCID>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.3 Artist (Artist)

This element specifies the artist of the source. Typically, this field is used in the Art and Sound Recording source types. [*Example*:

 ©ISO/IEC 2016 – All rights reserved

```
<b:Author>
  <b:Artist>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Artist>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.4 Author (Contributors List)

This element specifies the contributors to the source. [*Example*:

```
<b:Author>
  <b:Author>
    <b:NameList>
      <b:Person>
        <b:Last>Rothschiller</b:Last>
        <b:First>Chad</b:First>
      </b:Person>
    </b:NameList>
  </b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Rex</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <b:Translator>
      <b:NameList>
        <b:Person>
          <b:Last>Davis</b:Last>
          <b:First>Tristan</b:First>
        </b:Person>
      </b:NameList>
    </b:Translator>
  </b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AuthorType) is located in §A.6.6. *end note*]

22.6.2.5 Author (Author)

This element specifies the author of the source. [*Example*:

```
  <b:Author>
    <b:Author>
      <b:NameList>
        <b:Person>
          <b:Last>Jones</b:Last>
          <b:First>Brian</b:First>
        </b:Person>
      </b:NameList>
    </b:Author>
  </b:Author>
```

end example]

Attributes	Description
xml:space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by §2.10 of the XML 1.0 specification.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameOrCorporateType) is located in §A.6.6. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.6 BookAuthor (Book Author)

This element specifies the author of a book, when the primary author has authored the book section. For example, if person X writes a chapter in a book by person Y, person X is the Author and person Y is the BookAuthor. [*Example*:

```
<b:Author>
  <b:BookAuthor>
    <b:NameList>
      <b:Person>
        <b:Last>Rothschiller</b:Last>
        <b:First>Chad</b:First>
      </b:Person>
    </b:NameList>
  </b:BookAuthor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.7 BookTitle (Book Title)

This element specifies the title of a book, when the source is a book section. In this case, the title of the book section is the primary title. For example, if X is the title of a chapter in a book entitled , X is the Title and Y is the BookTitle. [*Example*:

```
<b:Source>
   …
  <b:Year>1992</b:Year>
  <b:City>Paris</b:City>
  <b:Publisher>Publisher</b:Publisher>
  <b:Pages>51-84</b:Pages>
  <b:Comments>Comments</b:Comments>
  <b:BookTitle>Book Title</b:BookTitle>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.6.2.8 Broadcaster (Broadcaster)

This element specifies the broadcaster of a source. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State or Province</b:StateProvince>
  <b:CountryRegion>Country or Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.9 BroadcastTitle (Broadcast Title)

This element specifies the broadcast title of a source. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State/Province</b:StateProvince>
  <b:CountryRegion>Country/Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.10 CaseNumber (Case Number)

This element specifies the case number of a source. Typically, this field is used in the Case source type. [*Example*:

```
…
<b:Title>Title (Case)</b:Title>
<b:Year>Year</b:Year>
<b:City>Place Published</b:City>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:CaseNumber>Case Number</b:CaseNumber>
<b:Court>Court</b:Court>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.11 ChapterNumber (Chapter Number)

This element specifies the number or index of the chapter being referenced. [*Example*:

```
…
<b:BookTitle>Title</b:BookTitle>
<b:Pages>23-65</b:Pages>
<b:Comments>Comments</b:Comments>
<b:ChapterNumber>6</b:ChapterNumber>
<b:RefOrder>1</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.12 City (City)

This element specifies the city in which the source was published, printed, or manufactured. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
…
<b:Title>Title</b:Title>
<b:Year>1997</b:Year>
<b:City>London</b:City>
<b:Publisher>Publihser</b:Publisher>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.13 Comments (Comments)

This element specifies any additional comments about the source. The documentation style determines whether the comments appear in the bibliography. [*Example*:

```
…
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Comments>Comments</b:Comments>
<b:RefOrder>2</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.14 Compiler (Compiler)

This element specifies the person who compiled the information in a source. [*Example*:

```
<b:Author>
  <b:Compiler>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Compiler>
</b:Author>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.15 Composer (Composer)

This element specifies the composer of a sound recording. [*Example*:

```
<b:Author>
  <b:Composer>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Composer>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.16 Conductor (Conductor)

This element specifies the conductor of a source. Typically, this field is used in the sound recording source type. [*Example*:

```
<b:Author>
  <b:Conductor>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Conductor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.17 ConferenceName (Conference or Proceedings Name)

This element specifies the title of the proceedings from a conference. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
…
<b:Comments>Comments</b:Comments>
<b:ConferenceName>Conference Name</b:ConferenceName>
<b:RefOrder>9</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.18 Corporate (Corporate Author)

This element specifies the corporate author, performer, or any field that can be a name. The element is used when an organization, rather than a person, is used. [*Example*:

```
<b:Author>
  <b:Author>
    <b:Corporate>Corporate Author</b:Corporate>
  </b:Author>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.19 Counsel (Counsel)

This element specifies the counsel, attorney, or attorneys in a case.

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.20 CountryRegion (Country or Region)

This element specifies the country or region of a source. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    <b:Title>Title (Interview)</b:Title>
    <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
    <b:StateProvince>State or Province</b:StateProvince>
    <b:CountryRegion>Country or Region</b:CountryRegion>
  </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.21 Court (Court)

This element specifies the court in which the case was presented. [*Example*:

```
…
<b:Year>1972</b:Year>
<b:CaseNumber>339 1018</b:CaseNumber>
<b:Court>Supreme Court</b:Court>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.22 Day (Day)

This element specifies the day on which a source was created or published. [*Example*:

```
…
<b:PeriodicalTitle>Periodical Title</b:PeriodicalTitle>
<b:Month>November</b:Month>
<b:Day>10</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.6.2.23 DayAccessed (Day Accessed)

This element specifies the day of the month a source was accessed. [*Example*:

```
<b:MonthAccessed>October</b:MonthAccessed>
<b:DayAccessed>5</b:DayAccessed>
<b:YearAccessed>2000</b:YearAccessed>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.24 Department (Department)

This element specifies the department in which a source originated, or to which a source was submitted. Typically, this field is used in the Report source type, which includes theses and dissertations. [*Example*:

```
…
<b:Institution>Harvard University</b:Institution>
<b:ThesisType>Doctoral Dissertation</b:ThesisType>
<b:Department>Department of Mathematics</b:Department>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.25 Director (Director)

This element specifies the director of a source. Typically, this field is used in the Film source type. [*Example*:

```
<b:Author>
  <b:Director>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Director>
</b:Author>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.26 Distributor (Distributor)

This element specifies the distributor of a source. Typically, this field is used in the Performance and Film source types. [*Example*:

```
…
<b:Distributor>Distributor</b:Distributor>
<b:Country>United States</b:Country>
<b:RefOrder>19</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.27 Edition (Editor)

This element specifies the edition of a source. [*Example*:

```
…
<b:Pages>1-34</b:Pages>
<b:Edition>Edition</b:Edition>
<b:Issue>Issue</b:Issue>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.28 Editor (Editor)

This element specifies the editor of a source. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<b:Author>
      <b:Editor>
        <b:NameList>
          <b:Person>
            <b:Last>Jaeschke</b:Last>
            <b:First>Rex</b:First>
          </b:Person>
        </b:NameList>
      </b:Editor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.29 First (Person's First, or Given, Name)

This element specifies a person's first name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Rex</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.30 Guid (GUID)

This element specifies the GUID of a source. [*Example*:

```
<b:Source>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
    …
    <b:RefOrder>2</b:RefOrder>
    <b:Guid>{EE06CBFE-1989-4533-A274-D81DFA436D79}</b:Guid>
    <b:LCID>0</b:LCID>
  </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.31 Institution (Institution)

This element specifies the institution of the source. Typically, this field is used in the Report source type, where it signifies the university or institute, and in the Art source type, where it signifies the museum or institution where the art is housed. [*Example*:

```
  <b:Source>
    …
    <b:Institution>Harvard University</b:Institution>
    <b:ThesisType>Dissertation</b:ThesisType>
    <b:RefOrder>12</b:RefOrder>
    <b:Guid>{6CB80970-81D3-476D-90D5-5C9D64E77FAF}</b:Guid>
    <b:LCID>0</b:LCID>
  </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.32 InternetSiteTitle (Internet Site Title)

This element specifies the title of an internet site. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.33 Interviewee (Interviewee)

This element specifies the person being interviewed. Typically, this field is used in the Interview source type. [*Example*:

```
<b:Interviewee>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:Interviewee>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.34 Interviewer (Interviewer)

This element specifies the person conducting an interview. Typically, this field is used in the Interview source type. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<b:Interviewer>
  <b:NameList>
    <b:Person>
      <b:Last>Davis</b:Last>
      <b:First>Tristan</b:First>
     </b:Person>
  </b:NameList>
</b:Interviewer>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.35 Inventor (Inventor)

This element specifies the inventor of a source. Typically, this field is used in the Patent source type. [*Example*:

```
<b:Author>
  <b:Inventor>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Inventor>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.36 Issue (Issue)

This element specifies the issue of a source. Typically, this field is used in the Journal Article and Article in Periodical source types. [*Example*:

```
<b:Source>
   …
  <b:Edition>Edition</b:Edition>
  <b:Issue>Issue</b:Issue>
  <b:RefOrder>28</b:RefOrder>
</b:Source>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.37 JournalName (Journal Name)

This element specifies the name of the journal. Typically, this field is used in the Journal Article source type. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.38 Last (Person's Last, or Family, Name)

This element specifies a person's last name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Jaeschke</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.39 LCID (Locale ID)

This element specifies the locale ID of a source, representing the source's language. The set of locale IDs shall be as specified in §22.9.2.6. [*Example*:

```
<b:LCID>ja-JP</b:LCID>
```

end example]

The possible values for this element are defined by the ST_Lang simple type (§22.9.2.6).

[*Note*: The W3C XML Schema definition of this element's content model (ST_Lang) is located in §A.6.9. *end note*]

22.6.2.40 Medium (Medium)

This element specifies the medium on or in which a source was created. Typically, this field is used in the Electronic source, sound recording, and film source types. [*Example*:

```
<b:Source>
  …
  <b:LCID>0</b:LCID>
  <b:Medium>DVD</b:Medium>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.41 Middle (Person's Middle, or Other, Name)

This element specifies a person's middle name. [*Example*:

```
<b:Author>
  <b:Editor>
    <b:NameList>
      <b:Person>
        <b:Last>Villaron</b:Last>
        <b:First>Shawn</b:First>
        <b:Middle>Alan</b:Middle>
      </b:Person>
    </b:NameList>
  </b:Editor>
</b:Author>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.42 Month (Month)

This element specifies the month in which a source was created or published. [*Example*:

```
…
<b:PeriodicalTitle>Time</b:PeriodicalTitle>
<b:Month>November</b:Month>
<b:Day>10</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.43 MonthAccessed (Month Accessed)

This element specifies the month during which the source was accessed. [*Example*:

```
…
<b:MonthAccessed>October</b:MonthAccessed>
<b:DayAccessed>5</b:DayAccessed>
<b:YearAccessed>2000</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.44 NameList (Name List)

This element specifies a list containing one or more names of a type of contributor to a source, such as a list of authors, editors, or translators. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<b:Author>
  <b:Author>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Author>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameListType) is located in §A.6.6. *end note*]

22.6.2.45 NumberVolumes (Number of Volumes)

This element specifies the number of volumes a source contains. [*Example*:

```
<b:Source>
   …
  <b:NumberVolumes>10</b:NumberVolumes>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.46 Pages (Pages)

This element specifies the page range being cited in a source. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.47 PatentNumber (Patent Number)

This element specifies the patent number of a source. Typically, this field is used in the Patent source type. [*Example*:

```
<b:Source>
  …
  <b:PatentNumber>1,000,000</b:PatentNumber>
  <b:RefOrder>26</b:RefOrder>
  <b:Guid>{8295ABC5-2DFD-4FA7-A2A7-A748917C1755}</b:Guid>
  <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.48 Performer (Performer)

This element specifies the performer. Typically, this field is used in the sound recording, performance, and film source types. [*Example*:

```
<b:Performer>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:Performer>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameOrCorporateType) is located in §A.6.6. *end note*]

22.6.2.49 PeriodicalTitle (Periodical Title)

This element specifies the title of a periodical. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
…
<b:PeriodicalTitle>Periodical Title</b:PeriodicalTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.50 Person (Person)

This element specifies a person who contributed to a source. [*Example*:

```
<b:Author>
  <b:NameList>
    <b:Person>
      <b:Last>Villaron</b:Last>
      <b:First>Shawn</b:First>
      <b:Middle>Alan</b:Middle>
    </b:Person>
  </b:NameList>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_PersonType) is located in §A.6.6. *end note*]

22.6.2.51 ProducerName (Producer Name)

This element specifies the person who produced a source. Typically, this field is used in the Internet site, Doc from internet site, Electronic source, Sound recording, Performance, and Film source types. [*Example*:

```
<b:ProducerName>
  <b:NameList>
    <b:Person>
      <b:Last>Rothschiller</b:Last>
      <b:First>Chad</b:First>
    </b:Person>
  </b:NameList>
</b:ProducerName>
```

end example]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.52 ProductionCompany (Production Company)

This element specifies the company that produced a source. Typically, this field is used in the Internet site, Document from internet site, Electronic source, Sound recording, Performance, and Film source types. [*Example*:

```
<b:Source>
   …
  <b:City>Chicago</b:City>
  <b:ProductionCompany>Production Company</b:ProductionCompany>
  <b:Medium>CD</b:Medium>
  <b:RefOrder>16</b:RefOrder>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.53 PublicationTitle (Publication Title)

This element specifies the title of the publication that contains the source. Typically, this field is used in the Electronic Source source type. [*Example*:

```
…
<b:Volume>Volume</b:Volume>
<b:PublicationTitle>Publication Title</b:PublicationTitle>
<b:Month>June</b:Month>
<b:Day>2</b:Day>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.54 Publisher (Publisher)

This element specifies the publisher of a source.

[*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
…
<b:City>London</b:City>
<b:Publisher>Publisher</b:Publisher>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>Volume</b:Volume>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.55 RecordingNumber (Recording Number)

This element specifies the recording number of a source. Typically, this field is used in the sound recording source type.

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.56 RefOrder (Reference Order)

This element specifies the reference order of a source. [*Example*:

```
<b:Source>
   …
  <b:City>Chicago</b:City>
  <b:ProductionCompany>Production Company</b:ProductionCompany>
  <b:Medium>CD</b:Medium>
  <b:RefOrder>16</b:RefOrder>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.57 Reporter (Reporter)

This element specifies the reporter of a source. Typically, this field is used in the Case source type. [*Example*:

```
<b:Source>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    …
    <b:Reporter>Reporter</b:Reporter>
    <b:RefOrder>27</b:RefOrder>
    <b:Guid>{CE314AB7-E824-4D10-B295-044C68EBED27}</b:Guid>
    <b:LCID>0</b:LCID>
  </b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.58 ShortTitle (Short Title)

This element specifies the short title of a source. [*Example*:

```
  …
  <b:City>London</b:City>
  <b:Publisher>Publisher</b:Publisher>
  <b:ShortTitle>Short Title</b:ShortTitle>
  <b:Volume>Volume</b:Volume>
  …
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.59 Source (Source)

This element specifies the bibliography entry for a source or reference work. [*Example*:

```
  <b:Source>
    …
    <b:NumberVolumes>10</b:NumberVolumes>
    <b:Comments>Comments</b:Comments>
  </b:Source>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SourceType) is located in §A.6.6. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.60 Sources (Sources)

This element specifies the sources in a collection.

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>SelectedStyle
(Selected Style)</td><td>Specifies the filename of a file which can be used to format the bibliographies and citations within this document.

If this file is of an unknown form or cannot be located, then the other attributes on this element can be used to determine the format to use.

[Example:

<code><b:Sources SelectedStyle="\APA.XSL" StyleName="APA"
URI="http://purl.oclc.org/ooxml/bibliographicStyle/APA"></code>

end example]

The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).</td></tr>
<tr><td>StyleName
(Documentation Style Name)</td><td>Specifies the name of the documentation style in which the bibliography and citations are formatted.

The following values shall be well-defined:

<table>
<tr><th>Value</th><th>Reference</th></tr>
<tr><td>APA</td><td>American Psychological Association. http://www.apa.org/. Publication Manual of the American Psychological Association, Fifth Edition.</td></tr>
<tr><td>Chicago</td><td>Chicago Manual of Style, 15th Edition
GB7714: GB7714-1987, Standardization Administration of China, 1987-05-05 (http://www.sac.gov.cn)</td></tr>
<tr><td>GOST - Name Soft</td><td>ГОСТ 7.1-2003 (GOST 7.1-2003) - The Federal Agency of the Russian Federation on Technical Regulating and Metrology - published by ИПК Издательство стандартов 2004</td></tr>
<tr><td>GOST - Title Sort</td><td>ГОСТ 7.1-2003 (GOST 7.1-2003) - The Federal Agency of the Russian Federation on Technical Regulating and Metrology - published by ИПК Издательство стандартов 2004</td></tr>
<tr><td>ISO 690 - First Element and Date</td><td>ISO 690-1987(E)-International Organization for Standardization-Second Edition 1987-08-15 (http://www.iso.org)
ISO 690-2:1997(E)-International Organization for Standardization-First Edition 1997-11-15 (http://www.iso.org)</td></tr>
<tr><td>ISO 690 - Numerical Reference</td><td>ISO 690-1987(E)-International Organization for Standardization-Second Edition 1987-08-15 (http://www.iso.org)
ISO 690-2:1997(E)-International Organization for Standardization-</td></tr>
</table>
</td></tr>
</table>

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>First Edition 1997-11-15 (http://www.iso.org)</td></tr>
<tr><td>MLA</td><td>Modern Language Association. http://www.mla.org/. MLA Handbook for Writers of Research Papers, Sixth Edition.</td></tr>
<tr><td>SIST02</td><td>Standard for Information of Science and Technology by Japan Science and Technology Agency, 2003(http://www.jst.go.jp/SIST/handbook/sist02sup/index.htm.</td></tr>
<tr><td>Turabian</td><td>A Manual for Writers of Term Papers, Theses, and Dissertations (Chicago Guides to Writing, Editing, and Publishing), by Kate L. Turabian, 1996.</td></tr>
<tr><td>Any other value</td><td>Implementation-defined.</td></tr>
</table>
[Example:
<pre><b:Sources SelectedStyle="\APA.XSL" StyleName="APA" URI="123"></pre>
end example]
The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
</td></tr>
<tr><td>URI (Uniform Resource Identifier)</td><td>
Specifies a URI or unique identifier with which a documentation style is associated; can be used to uniquely identify versions of styles that share a StyleName.
[Example:
<pre><b:Sources SelectedStyle="\APA.XSL" StyleName="APA" URI="123"></pre>
end example]
The possible values for this attribute are defined by the ST_String simple type (§22.9.2.13).
</td></tr>
</table>

[*Note*: The W3C XML Schema definition of this element's content model (CT_Sources) is located in §A.6.6. *end note*]

22.6.2.61 SourceType (Source Type)

This element specifies the type of source being cited.

The possible values for this element are defined by the ST_SourceType simple type (§22.6.3.1).

[*Note*: The W3C XML Schema definition of this element's content model (ST_SourceType) is located in §A.6.6. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.62 StandardNumber (Standard Number)

This element specifies the standard number, such as ISBN or ISSN, of a source. [*Example*:

```
<b:Source>
   …
  <b:NumberVolumes>10</b:NumberVolumes>
  <b:StandardNumber>ISBN or ISSN</b:StandardNumber>
  <b:Comments>Comments</b:Comments>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.63 StateProvince (State or Province)

This element specifies the state or province in which a source was created or published. [*Example*:

```
<b:Source>
   …
  <b:ProgramTitle>Program Title</b:ProgramTitle>
  <b:Broadcaster>Broadcaster</b:Broadcaster>
  <b:Station>Station</b:Station>
  <b:RefOrder>1</b:RefOrder>
  <b:Title>Title (Interview)</b:Title>
  <b:BroadcastTitle>Broadcast Title</b:BroadcastTitle>
  <b:StateProvince>State/Province</b:StateProvince>
  <b:CountryRegion>Country/Region</b:CountryRegion>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.64 Station (Station)

This element specifies the station on which an interview was broadcasted. Typically, this field is used in the Interview source type. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
…
<b:Month>November</b:Month>
<b:Day>18</b:Day>
<b:Broadcaster>ABC</b:Broadcaster>
<b:Station>WABC</b:Station>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.65 Tag (Tag)

This element specifies the tag name of a source. [*Example*:

```
<b:Source>
  <b:Tag>New01</b:Tag>
  …
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.66 Theater (Theater)

This element specifies the theater in which a source was performed or viewed. Typically, this field is used in the Performer source type. [*Example*:

```
…
<b:Theater>Theater Name</b:Theater>
<b:Month>October</b:Month>
<b:Day>25</b:Day>
<b:RefOrder>19</b:RefOrder>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.6.2.67 ThesisType (Thesis Type)

This element specifies the type of report being cited, such as Thesis, Dissertation, or Book Report. Typically, this field is used in the Report source type. [*Example*:

```
…
<b:Institution>Harvard University</b:Institution>
<b:ThesisType>Doctoral Dissertation</b:ThesisType>
<b:Department>Department of Mathematics</b:Department>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.68 Title (Title)

This element specifies the title of a source. [*Example*:

```
…
</b:Author>
<b:Title>Title</b:Title>
<b:Year>2005</b:Year>
<b:City>Seattle</b:City>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.69 Translator (Translator)

This element specifies the translator of a source. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
<b:Author>
  <b:Translator>
    <b:NameList>
      <b:Person>
        <b:Last>Davis</b:Last>
        <b:First>Tristan</b:First>
      </b:Person>
    </b:NameList>
  </b:Translator>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_NameType) is located in §A.6.6. *end note*]

22.6.2.70 Type (Patent Type)

This element specifies the type of patent. Typically, this field is used in the Patent source type.

[*Example*:

```
<Source>
  <b:Type>Patent Type</b:Type>
  <b:Guid>{8295ABC5-2DFD-4FA7-A2A7-A748917C1755}</b:Guid>
  <b:LCID>0</b:LCID>
</Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.71 URL (URL)

This element specifies the URL of the source. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

```
…
<b:MonthAccessed>September</b:MonthAccessed>
<b:DayAccessed>1</b:DayAccessed>
<b:YearAccessed>1998</b:YearAccessed>
<b:URL>URL</b:URL>
<b:RefOrder>29</b:RefOrder>
…
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.72 Version (Version)

This element specifies the version of the source. Typically, this field is used in the Internet Site and Document from Internet Site source types. [*Example*:

```
<b:Source>
    …
  <b:Version>3.0</b:Version>
  <b:RefOrder>31</b:RefOrder>
  <b:Guid>{F06D8D48-7FD7-4515-88E9-EC70AB9BE792}</b:Guid>
  <b:LCID>0</b:LCID>
</b:Source>
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.73 Volume (Volume)

This element specifies the volume of the source. [*Example*:

```
…
<b:Title>Article Title</b:Title>
<b:Year>2000</b:Year>
<b:ShortTitle>Short Title</b:ShortTitle>
<b:Volume>100</b:Volume>
<b:Comments>Comments</b:Comments>
<b:JournalName>Journal Name</b:JournalName>
<b:Pages>91-160</b:Pages>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.6.2.74 Writer (Writer)

This element specifies the writer of the source. Typically, this field is used in the Performance and Film source types. [*Example*:

```
<b:Author>
  <b:Writer>
    <b:NameList>
      <b:Person>
        <b:Last>Jones</b:Last>
        <b:First>Brian</b:First>
      </b:Person>
    </b:NameList>
  </b:Writer>
</b:Author>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (ST_NameType) is located in §A.6.6. *end note*]

22.6.2.75 Year (Year)

This element specifies the year in which a source was created or published.

[*Example*:

```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.2.76 YearAccessed (Year Accessed)

This element specifies the month during which the source was accessed. [*Example*:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
…
<b:Title>Title</b:Title>
<b:InternetSiteTitle>Internet Site Title</b:InternetSiteTitle>
<b:Month>July</b:Month>
<b:Day>1</b:Day>
<b:Year>2001</b:Year>
<b:MonthAccessed>Sept.</b:MonthAccessed>
<b:DayAccessed>22</b:DayAccessed>
<b:YearAccessed>1999</b:YearAccessed>
…
```

end example]

The possible values for this element are defined by the ST_String simple type (§22.9.2.13).

[*Note*: The W3C XML Schema definition of this element's content model (ST_String) is located in §A.6.9. *end note*]

22.6.3 Simple Types

This is the complete list of simple types dedicated to Bibliography.

22.6.3.1 ST_SourceType (Bibliographic Data Source Types)

This simple type specifies the possible types of sources that can be used within bibliographic data in an Office Open XML document.

[*Note:* The Office Open XML formats support a collection of predefined source types based on the categories most commonly used in various citation and bibliography style guidelines . The set of predefined source types can be extended as needed. The recommended approach for extending this set is to use the `Misc` type, and then leverage the methods described in ISO/IEC 29500-3 for extending the format with new attributes or elements. *end note*]

This simple type's contents are a restriction of the ST_String datatype (§22.9.2.13).

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
Art (Art)	Art
ArticleInAPeriodical (Article in a Periodical)	Article in a Periodical
Book (Book)	Book
BookSection (Book Section)	Book Section
Case (Case)	Case
ConferenceProceedings (Conference Proceedings)	Conference Proceedings
DocumentFromInternetSite (Document from Internet Site)	Document from Internet Site

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Enumeration Value	Description
ElectronicSource (Electronic Source)	Electronic Source
Film (Film)	Film
InternetSite (Internet Site)	Internet Site
Interview (Interview)	Interview
JournalArticle (Journal Article)	Journal Article
Misc (Miscellaneous)	Miscellaneous
Patent (Patent)	Patent
Performance (Performance)	Performance
Report (Reporter)	Report
SoundRecording (Sound Recording)	Sound Recording

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_SourceType) is located in §A.6.6. *end note*]

22.7 Additional Characteristics

In order to allow producers of Office Open XML to describe specific contextual conditions under which the document was created, additional characteristics can be provided within the Additional Characteristics part using the syntax defined below.

The set of additional characteristics is designed to be an extensible list, and can provide a consumer with more information on how to interpret the file. ISO/IEC 29500 defines one set of characteristics; however, additional grammars can be created and associated with a unique URI via the vocabulary attribute.

22.7.1 Table of Contents

This subclause is informative.

22.7.2 Elements 3780
22.7.2.1 additionalCharacteristics (Set of Additional Characteristics) 3781
22.7.2.2 characteristic (Single Characteristic) 3781
22.7.3 Simple Types 3783
22.7.3.1 ST_Relation (Characteristic Relationship Types) 3783

End of informative text.

22.7.2 Elements

The following elements define the contents of the Additional Characteristics schema:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.7.2.1 additionalCharacteristics (Set of Additional Characteristics)

This element is the root element of the Additional Characteristics part and contains the list of additional characteristics for an Office Open XML document.

[*Example:* The following content in an Additional Characteristics part would specify that the producing spreadsheet application supports from 0 to 10,000 columns, and that column ranges should be interpreted accordingly:

```
<additionalCharacteristics>
  <characteristic name="numColumns" relation="le" val="10000"/>
  <characteristic name="numColumns" relation="ge" val="0"/>
</additionalCharacteristics>
```

end example]

[*Note*: The W3C XML Schema definition of this element's content model (CT_AdditionalCharacteristics) is located in §A.6.7. *end note*]

22.7.2.2 characteristic (Single Characteristic)

This element specifies a single characteristic. The type of characteristic is defined by the name attribute.

[*Example:* A producer can inform the consumer that the computations used to calculate the stored numbers in the formulas belong to a value space expressed by ranges of the binary mantissa and exponent. A consumer can optionally check those characteristics to determine whether, for example, the values should be recalculated. The XML for this would be:

```
<additionalCharacteristics>
  <characteristic name='precisionMantissa' relation='gt'
     val='-9007199254740992'/>
  <characteristic name='precisionMantissa' relation='lt'
     val='9007199254740992'/>
  <characteristic name='precisionExponent' relation='ge' val='-1075'/>
  <characteristic name='precisionExponent' relation='le' val='970'/>
</additionalCharacteristics>
```

end example]

Attributes	Description
name (Name of Characteristic)	Specifies the name of the characteristic. There are no constraints on the value of the name attribute, but each name shall be associated with a specific vocabulary via the vocabulary attribute. The values defined by ISO/IEC 29500 shall be associated with a null vocabulary value, and are as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Attributes	Description

Name Value	Property Specified
numColumns	Number of Columns supported by the spreadsheet producer.
numRows	Number of Rows supported by the spreadsheet producer.
functionVersion	Version of the function specification used
precisionMantissa	Allowed values of the mantissa of numbers within spreadsheet cells/formulas when expressed in base 2.
precisionExponent	Allowed values of the exponent of numbers within spreadsheet cells/formulas when expressed in base 2.
numWorkbookColors	Number of Workbook colors
numConditionalFormatConditions	Number of condition format conditions on a workbook cell
nummaxSortLevels	Number of level of sorting on a range or table
numAutoFilterItems	Number of items shown in the Auto-filter dropdown
numDisplayCellChars	Number of characters that can display in a cell
numPrintCellChars	Number of characters per cell that Excel can print
numUnqiueCellStyles	Number of unique cell styles in a workbook (combinations of all cell formatting)
numFormulaLengthChars	Length of formulas in characters
numFormulaNestingLevel	Number of levels of formula nesting
numFunctionArguments	Number of arguments to a function
numPivotTableRows	Number of rows in a pivot table
numPivotTableColumns	Number of columns in a pivot table
numUniquePivotFieldItems	Number of unique items in a pivot field
numPivotTableMDXNameChars	Number of characters in a MDX name for a pivot table item
numPivotTableRelationChars	String length for a relationship pivot table
numPivotTableFieldLabelChars	Length of field labels in PivotTable including caption length limitations
numPivotTableFields	Number of fields in a pivot table
numSheetXRefArrayFormulas	The number of array formulas in a worksheet that can refer to another (given) worksheet

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

relation (Relationship of Value to Name)	Specifies how the contents of the value attribute should be interpreted in the context of this characteristic.
	
	[*Example:* The following would specify that the application supports from 0 to 10,000 columns, and that column ranges should be interpreted accordingly:
	
	`<additionalCharacteristics>`
	`  <characteristic name="numColumns" relation="le" val="10000"/>`
	`  <characteristic name="numColumns" relation="ge" val="0"/>`
	`</additionalCharacteristics>`
	
	end example]
	
	The possible values for this attribute are defined by the ST_Relation simple type (§22.7.3.1).

val (Characteristic Value)	Specifies the value of the characteristic. The possible values for this attribute are defined by the W3C XML Schema string datatype.
vocabulary (Characteristic Grammar)	Specifies a URI defining the characteristic grammar with which the name attribute value shall be interpreted. If this attribute is omitted, then the default grammar (as defined above) shall be used. The possible values for this attribute are defined by the W3C XML Schema anyURI datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Characteristic) is located in §A.6.7. *end note*]

22.7.3 Simple Types

This is the complete list of simple types dedicated to Additional Characteristics.

22.7.3.1 ST_Relation (Characteristic Relationship Types)

This simple type specifies the possible relationships between a characteristic's name and value attributes.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
eq (Equal To)	Equal to.
ge (Greater Than or Equal to)	Greater than or equal to.
gt (Greater Than)	Greater than.
le (Less Than or Equal To)	Less than or equal to.
lt (Less Than)	Less than.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Relation) is located in §A.6.7. *end note*]

22.8 Office Document Relationships

Within an Office Open XML document, it is necessary to be able to explicitly reference one part within the package from another [*Example*: A PresentationML Slide needs to be able to explicitly reference each picture within it to know where each one is anchored. *end example*]

In order to ensure that all such explicit relationship references are easily identifiable within a document, all such relationships are included within attributes in this namespace. This namespace therefore only serves to define attributes used throughout Office Open XML to create explicit relationships, and a single simple type for such attributes.

22.8.1 Table of Contents

This subclause is informative.

22.8.2 Simple Types 3784
22.8.2.1 ST_RelationshipId (Explicit Relationship ID) 3784

End of informative text.

22.8.2 Simple Types

This is the complete list of simple types dedicated to Office Document Relationships.

22.8.2.1 ST_RelationshipId (Explicit Relationship ID)

This simple type specifies the relationship ID in a part's relationship item which is the target of an explicit relationship from the parent XML element.

The kind of relationship which shall be the target of the relationship specified shall be determined based on the context of the parent XML element.

[*Example*: Consider the following markup in an Office Open XML document:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<… r:id="rId5" />
```

The id attribute is of type ST_RelationshipID, and therefore the relationship with ID `rId5` must be the target of an explicit relationship from the source part, based on the context of the parent XML element. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_RelationshipId) is located in §A.6.8. *end note*]

22.9 Shared Simple Types

The following simple types represent common value formats used throughout Office Open XML, and have been centralized in order to ensure their usage remains consistent.

22.9.1 Table of Contents

This subclause is informative.

22.9.2 Simple Types 3785
22.9.2.1 ST_CalendarType (Calendar Types) 3786
22.9.2.2 ST_ConformanceClass (Document Conformance Class Value) 3787
22.9.2.3 ST_FixedPercentage (Fixed Percentage Value with Sign) 3788
22.9.2.4 ST_Guid (128-Bit GUID) 3788
22.9.2.5 ST_HexColorRGB (Hexadecimal Color Value) 3789
22.9.2.6 ST_Lang (Language Reference) 3790
22.9.2.7 ST_OnOff (On/Off Value) 3790
22.9.2.8 ST_Panose (Panose-1 Number) 3790
22.9.2.9 ST_Percentage (Percentage Value with Sign) 3791
22.9.2.10 ST_PositiveFixedPercentage (Positive Fixed Percentage Value with Sign) 3792
22.9.2.11 ST_PositivePercentage (Positive Percentage Value with Sign) 3792
22.9.2.12 ST_PositiveUniversalMeasure (Positive Universal Measurement) 3793
22.9.2.13 ST_String (String) 3793
22.9.2.14 ST_TwipsMeasure (Measurement in Twentieths of a Point) 3793
22.9.2.15 ST_UniversalMeasure (Universal Measurement) 3794
22.9.2.16 ST_UnsignedDecimalNumber (Unsigned Decimal Number Value) 3794
22.9.2.17 ST_VerticalAlignRun (Vertical Positioning Location) 3795
22.9.2.18 ST_XAlign (Horizontal Alignment Location) 3796
22.9.2.19 ST_Xstring (Escaped String) 3797
22.9.2.20 ST_YAlign (Vertical Alignment Location) 3797
22.9.2.21 ST_XmlName (XML Name) 3799

End of informative text.

22.9.2 Simple Types

This is the complete list of simple types dedicated to Shared Simple Types.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.9.2.1 ST_CalendarType (Calendar Types)

This simple type specifies the possible types of calendars which can be used within the context of an Office Open XML document.

[*Example*: Consider the following structured document tag properties:

```
<w:sdtPr>
  <w:date w:fullDate="2006-01-01T06:30:00Z">
    <w:calendar w:val="gregorian"/>
  </w:date>
</w:sdtPr>
```

The calendar element specifies that the calendar type for a calendar which can be displayed in the document must be the Gregorian calendar format (`gregorian`). *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
gregorian (Gregorian)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. This calendar should be localized into the appropriate language.
gregorianArabic (Gregorian Arabic Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in Arabic.
gregorianMeFrench (Gregorian Middle East French Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in Middle East French.
gregorianUs (Gregorian English Calendar)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be presented in English.
gregorianXlitEnglish (Gregorian Transliterated English)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be the representation of the English strings in the corresponding Arabic characters (the Arabic transliteration of the English for the Gregorian calendar).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
gregorianXlitFrench (Gregorian Transliterated French)	Specifies that the Gregorian calendar, as defined in ISO 8601, shall be used. The values for this calendar should be the representation of the French strings in the corresponding Arabic characters (the Arabic transliteration of the French for the Gregorian calendar).
hebrew (Hebrew)	Specifies that the Hebrew lunar calendar, as described by the Gauss formula for Passover [Har'El, Zvi] and The Complete Restatement of Oral Law (Mishneh Torah), shall be used.
hijri (Hijri)	Specifies that the Hijri lunar calendar, as described by the Kingdom of Saudi Arabia, Ministry of Islamic Affairs, Endowments, Da'wah and Guidance, shall be used.
japan (Japanese Emperor Era)	Specifies that the Japanese Emperor Era calendar, as described by Japanese Industrial Standard JIS X 0301, shall be used.
korea (Korean Tangun Era)	Specifies that the Korean Tangun Era calendar, as described by Korean Law Enactment No. 4, shall be used.
none (No Calendar Type)	Specifies that no calendar should be used.
saka (Saka Era)	Specifies that the Saka Era calendar, as described by the Calendar Reform Committee of India, as part of the Indian Ephemeris and Nautical Almanac, shall be used.
taiwan (Taiwan)	Specifies that the Taiwanese calendar, as defined by the Chinese National Standard CNS 7648, shall be used.
thai (Thai)	Specifies that the Thai calendar, as defined by the Royal Decree of H.M. King Vajiravudh (Rama VI) in Royal Gazette B. E. 2456 (1913 A.D.) and by the decree of Prime Minister Phibunsongkhram (1941 A.D.) to start the year on the Gregorian January 1 and to map year zero to Gregorian year 543 B.C., shall be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_CalendarType) is located in §A.6.9. *end note*]

22.9.2.2 ST_ConformanceClass (Document Conformance Class Value)

This simple type specifies the conformance class to which a particular Office Open XML document conforms.

[*Example*: Consider the following SpreadsheetML Workbook part markup:

```
<workbook … conformance="transitional">
  …
</workbook>
```

This SpreadsheetML document has a conformance attribute value of `transitional`, therefore it conforms to the SML Transitional conformance class. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
strict (Office Open XML Strict)	Specifies that the document conforms to Office Open XML Strict.
transitional (Office Open XML Transitional)	Specifies that the document conforms to Office Open XML Transitional.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ConformanceClass) is located in §A.6.9. *end note*]

22.9.2.3 ST_FixedPercentage (Fixed Percentage Value with Sign)

This simple type specifies that its contents will contain a percentage measurement from -100% up to and including 100%, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: -?((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_FixedPercentage) is located in §A.6.9. *end note*]

22.9.2.4 ST_Guid (128-Bit GUID)

This simple type specifies that its values shall be a 128-bit globally unique identifier (GUID) value.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider the following WordprocessingML fragment for the properties of a single glossary document entry:

```
<w:docPartPr>
  …
  <w:guid w:val="{00000000-5BD2-4BC8-9F70-7020E1357FB2}" />
  …
</w:docPartPr>
```

The guid element specifies that the unique identifier associated with the parent entry shall be {00000000-5BD2-4BC8-9F70-7020E1357FB2}. This value can be used as needed by an application, for example, to uniquely identify a part regardless of its name. *end example*]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: \{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Guid) is located in §A.6.9. *end note*]

22.9.2.5 ST_HexColorRGB (Hexadecimal Color Value)

This simple type specifies that its contents shall contain a color value in RRGGBB hexadecimal format, specified using six hexadecimal digits. Each of the red, green, and blue color values, from 0-255, is encoded as two hexadecimal digits.

[*Example*: Consider a color defined as follows:

```
Red:    122
Green:  23
Blue:   209
```

The resulting RRGGBB value would be 7A17D1, as each color is transformed into its hexadecimal equivalent. *end example*]

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 6 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_HexColorRGB) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

22.9.2.6 ST_Lang (Language Reference)

This simple type specifies that its contents contains a language identifier as defined by RFC 4646/BCP 47.

The contents of this language are interpreted based on the context of the parent XML element.

[*Example*: Consider a language code defined as follows :

```
<w:lang w:val="en-CA" />
```

This language is therefore specified as English (en) and Canada (CA), resulting in use of the English (Canada) language setting. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Lang) is located in §A.6.9. *end note*]

22.9.2.7 ST_OnOff (On/Off Value)

This simple type specifies a set of values for any binary (`true` or `false`) property defined in a WordprocessingML document.

A value of `1` or `true` specifies that the property shall be turned on. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.

A value of `0` or `false` specifies that the property shall be explicitly turned off.

This simple type's contents are a restriction of the W3C XML Schema boolean datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_OnOff) is located in §A.6.9. *end note*]

22.9.2.8 ST_Panose (Panose-1 Number)

This simple type specifies a Panose-1 font classification. This value is used as one piece of information to guide selection of a similar alternate font if the desired font is unavailable.

[*Guidance*: ISO/IEC 14496-22 refers to the Panose specification dated 1997, but some implementations of ISO/IEC 29500 may follow an older precursor version of the Panose specification. Some values between the two versions are incompatible and the handling of these differences is implementation-defined.

Panose-1 values designed for the older version should match one of the following 3 regular expressions:

```
\s*(00\s*){10}\s*
\s*(01\s*){10}\s*
\s*0[2-5]\s*0[0-9A-Fa-f]\s*0[0-9ABab]\s*0[0-9]\s*0[0-9]\s*0[0-9Aa]\s*0[0-
9ABab]\s*0[0-9A-Fa-f]\s*0[0-9A-Da-d]\s*0[0-7]\s*
```

Panose-1 values designed for ISO/IEC 14496-22 should match one of the following 6 regular expressions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
\s*(00\s*){10}\s*
\s*(01\s*){10}\s*
\s*02\s*0[0-9A-Fa-f]\s*0[0-9ABab]\s*0[0-9]\s*0[0-9]\s*0[0-9Aa]\s*0[0-9ABab]\s*0[0-
9A-Fa-f]\s*0[0-9A-Da-d]\s*0[0-7]\s*
\s*03\s*0[0-9]\s*0[0-9ABab]\s*0[0-3]\s*0[0-6]\s*0[0-9]\s*0[0-9Aa]\s*0[0-9A-Da-
d]\s*0[0-9A-Da-d]\s*0[0-6]\s*
\s*04\s*0[0-9A-Ca-c]\s*0[0-9ABab]\s*0[0-9]\s*0[0-9A-Da-d]\s*(0[0-9A-Fa-
f]|10)\s*0[0-7]\s*0[0-8]\s*0[0-9A-Fa-f]\s*0[0-5]\s*
\s*05\s*0[0-9A-Ca-c]\s*01\s*0[0-3]\s*01\s*0[0-9]\s*0[0-9]\s*0[0-9]\s*0[0-9]\s*0[0-
9]\s*
```

end guidance]

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  …
</w:font>
```

The panose1 element specifies its Panose-1 number via its val attribute value of `02020603050405020304`. *end example*]

This simple type's contents are a restriction of the W3C XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a length of exactly 20 hexadecimal digit(s).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Panose) is located in §A.6.9. *end note*]

22.9.2.9 ST_Percentage (Percentage Value with Sign)

This simple type specifies that its contents will contain a percentage measurement, with a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

- This simple type's contents shall match the following regular expression pattern: -?[0-9]+(\.[0-9]+)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Percentage) is located in §A.6.9. *end note*]

22.9.2.10 ST_PositiveFixedPercentage (Positive Fixed Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage measurement from 0% to 100% inclusive, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: ((100)|([0-9][0-9]?))(\.[0-9][0-9]?)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveFixedPercentage) is located in §A.6.9. *end note*]

22.9.2.11 ST_PositivePercentage (Positive Percentage Value with Sign)

This simple type specifies that its contents will contain a positive percentage measurement, including a trailing percent sign (U+0025).

[*Example*: Consider the following WordprocessingML fragment:

```
<w:tcPr>
  <w:tcW w:type="pct" w:w="33.3%" />
</w:pPr>
```

The value of the w attribute is the width of the associated table cell. *end example*]

This simple type's contents are a restriction of the ST_Percentage datatype (§22.9.2.9).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: [0-9]+(\.[0-9]+)?%.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositivePercentage) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

22.9.2.12 ST_PositiveUniversalMeasure (Positive Universal Measurement)

This simple type specifies that its contents will contain a measurement expressed using one of common measure units. The content of this type is a positive decimal number immediately followed by a unit identifier. Unit identifiers are case sensitive and shall be in lowercase. Conforming applications are not required to preserve units of measure between loading and saving a particular document.

The same set of units of measure as in ST_UniversalMeasure type (§22.9.2.15) shall be supported.

This simple type's contents are a restriction of the ST_UniversalMeasure datatype (§22.9.2.15).

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: [0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_PositiveUniversalMeasure) is located in §A.6.9. *end note*]

22.9.2.13 ST_String (String)

This simple type specifies that its contents contains a string.

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_String) is located in §A.6.9. *end note*]

22.9.2.14 ST_TwipsMeasure (Measurement in Twentieths of a Point)

This simple type specifies that its contents contain wither:

- A positive whole number, whose contents consist of a measurement in twentieths of a point (equivalent to 1/1440th of an inch), or
- A positive decimal number immediately following by a unit identifier.

The contents of this measurement are interpreted based on the context of the parent XML element.

[*Example*: Consider an attribute value of `720` whose type is ST_TwipsMeasure. This attribute value specifies a size of one-half of an inch or 36 points (`720` twentieths of a point = 36 points = 0.5 inches). *end example*]

[*Example*: Consider an attribute value of `12.7mm` whose type is ST_HpsMeasure. This attribute value specifies a size of 0.0127 meter or one-half of an inch or 36 points. *end example*]

This simple type is a union of the following types:

- The ST_PositiveUniversalMeasure simple type (§22.9.2.12).
- The ST_UnsignedDecimalNumber simple type (§22.9.2.16).

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_TwipsMeasure) is located in §A.6.9. *end note*]

22.9.2.15 ST_UniversalMeasure (Universal Measurement)

This simple type specifies that its contents will contain measurement expressed using one of common measure units. The content of this type is a decimal number immediately followed by a unit identifier. Unit identifiers are case sensitive and shall be in lowercase. Conforming applications are not required to preserve units of measure between loading and saving a particular document.

The following table lists units of measure which are allowed together with their definition based on existing standard or expressed as a conversion from other unit of measure.

Unit Identifier	Definition
`cm`	As defined in ISO 31.
`mm`	As defined in ISO 31.
`in`	1 in = 2.54 cm (informative)
`pt`	1 pt = 1/72 in (informative)
`pc`	1 pc = 12 pt (informative)
`pi`	1 pi = 12 pt (informative)

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents shall match the following regular expression pattern: -?[0-9]+(\.[0-9]+)?(mm|cm|in|pt|pc|pi).

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UniversalMeasure) is located in §A.6.9. *end note*]

22.9.2.16 ST_UnsignedDecimalNumber (Unsigned Decimal Number Value)

This simple type specifies that its contents contain a positive whole decimal number, whose contents are interpreted based on the context of the parent XML element.

[*Example*: Consider the following WordprocessingML fragment:

```
<w:pPr>
  <w:divId w:val="1512645511" />
</w:pPr>
```

The value of the val attribute is the ID of the associated HTML `div`.

However, consider the following fragment:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<w:ilvl w:val="1">
  …
</w:ilvl>
```

In this case, the decimal number in the val attribute is the ID of the associated numbering level. In each case, the decimal number value is interpreted in the context of the parent element. *end example*]

This simple type's contents are a restriction of the W3C XML Schema unsignedLong datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_UnsignedDecimalNumber) is located in §A.6.9. *end note*]

22.9.2.17 ST_VerticalAlignRun (Vertical Positioning Location)

This simple type specifies possible values for the alignment of the contents of this run in relation to the default appearance of the run's text. This allows the text to be repositioned as subscript or superscript without altering the font size of the run properties.

[*Example*: Consider a run which must be positioning as superscript when displaying its contents. This requirement would be specified using the following WordprocessingML:

```
<w:rPr>
  <w:vertAlign w:val="superscript" />
</w:rPr>
```

The resulting run is positioned as superscript, therefore it is rendered in a smaller size above the default baseline location for the contents of the run. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
baseline (Regular Vertical Positioning)	Specifies that the text in the parent run shall be located at the baseline and presented in the same size as surrounding text.
subscript (Subscript)	Specifies that this text should be subscript. This setting shall lower the text in this run below the baseline and change it to a smaller size, if a smaller size is available.
superscript (Superscript)	Specifies that this text should be superscript. This setting shall raise the text in this run above the baseline and change it to a smaller size, if a smaller size is available.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_VerticalAlignRun) is located in §A.6.9. *end note*]

22.9.2.18 ST_XAlign (Horizontal Alignment Location)

This simple type specifies the set of possible relative horizontal positions for the parent floating object. This relative position is specified relative to the horizontal anchor specified by the parent object.

[*Example*: Consider the following WordprocessingML fragment specifying a text frame:

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187"
w:wrap="around" w:vAnchor="text" w:hAnchor="page" w:xAlign="left" w:y="73" />
  </w:pPr>
  <w:r>
    <w:t>Text Frame Content.</w:t>
  </w:r>
</w:p>
```

This text frame specifies by the presence of the xAlign attribute to align the frame on the `left` side of the anchor object, in this case, the page. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
center (Centered Horizontally)	Specifies that the parent object shall be centered with respect to the anchor settings. [*Example*: Centered on the page horizontally. *end example*]
inside (Inside)	Specifies that the parent object shall be inside of the anchor object. [*Example*: Inside the text margin horizontally. *end example*]
left (Left Aligned Horizontally)	Specifies that the parent object shall be left aligned with respect to the anchor settings. [*Example*: Left aligned on the page horizontally. *end example*]
outside (Outside)	Specifies that the parent object shall be outside of the anchor object. [*Example*: Outside the text margin horizontally. *end*

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

Enumeration Value	Description
	example]
right (Right Aligned Horizontally)	Specifies that the parent object shall be right aligned with respect to the anchor settings. [*Example*: Right aligned on the page horizontally. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_XAlign) is located in §A.6.9. *end note*]

22.9.2.19 ST_Xstring (Escaped String)

String of characters with support for escaped invalid-XML characters.

For all characters which cannot be represented in XML as defined by the XML 1.0 specification, the characters are escaped using the Unicode numerical character representation escape character format _xHHHH_, where H represents a hexadecimal character in the character's value. [*Example*: The Unicode character 8 is not permitted in an XML 1.0 document, so it must be escaped as _x0008_. *end example*]

This simple type's contents are a restriction of the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_Xstring) is located in §A.6.9. *end note*]

22.9.2.20 ST_YAlign (Vertical Alignment Location)

This simple type specifies the set of possible relative vertical positions for the parent floating object. This relative position is specified relative to the vertical anchor specified by the parent object.

[*Example*: Consider the following WordprocessingML fragment specifying a text frame:

```
<w:p>
  <w:pPr>
    <w:framePr w:w="2419" w:h="2189" w:hRule="atLeast" w:hSpace="187"
w:wrap="around" w:vAnchor="margin" w:hAnchor="page" w:x="1643" w:y="73"
w:yAlign="center" />
  </w:pPr>
  <w:r>
    <w:t>Text Frame Content.</w:t>
  </w:r>
</w:p>
```

This text frame specifies by the presence of the yAlign attribute to vertically align the frame in the `center` of the anchor object, in this case, the `margin`. *end example*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

This simple type's contents are a restriction of the W3C XML Schema string datatype.

This simple type is restricted to the values listed in the following table:

Enumeration Value	Description
bottom (Bottom)	Specifies that the parent object shall be vertically aligned to the bottom edge of the anchor object. [*Example*: At the bottom of the current paragraph. *end example*]
center (Centered Vertically)	Specifies that the parent object shall be vertically centered with respect to the anchor object. Shall not be used with the baseJc element. [*Example*: Centered on the page vertically. *end example*]
inline (In line With Text)	Specifies that the parent object shall be vertically aligned in line with the surrounding text (i.e. shall not allow any text wrapping around it when positioned in the document. Shall not be used with the baseJc element.
inside (Inside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned inside that object. Shall not be used with the baseJc element. [*Example*: Inside the text margins vertically. *end example*]
outside (Outside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned outside that object. Shall not be used with the baseJc element. [*Example*: Outside the text margins vertically. *end example*]
top (Top)	Specifies that the parent object shall be vertically aligned to the top edge of the anchor object . [*Example*: At the top of the current paragraph. *end example*]

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_YAlign) is located in §A.6.9. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

22.9.2.21 ST_XmlName (XML Name)

This simple type shall contain an XML non-colonized name (NCName).

[*Example*: Consider the following WordprocessingML fragment for the properties of a single custom XML element:

```
<w:customXmlPr>
<w:attr w:name="company"  w:uri="http://schemas.openxmlformats.org/2006/example"
… />
</w: customXmlPr>
```

The attr element specifies that the NCName associated with the attribute name shall be company. *end example*]

This simple type's contents are a restriction of the W3C XML Schema NCName datatype.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

23. Custom XML Schema References

This namespace defines the set of properties which define the location and properties associated with one or more custom XML schemas which have been stored within the contents of a Office Open XML document. Collectively, the set of schemas associated with a document's custom XML markup are referred to as that document's *schema library*. The schema library then stores the set of unique XML namespaces used within the document's custom XML markup, and allows applications to 'tag' these namespaces with appropriate metadata.

23.1 Table of Contents

This subclause is informative.

23.2 Elements 3800
23.2.1 schema (Custom XML Schema Reference) 3800
23.2.2 schemaLibrary (Embedded Custom XML Schema Supplementary Data) 3802

End of informative text.

23.2 Elements

The following information describes the elements in this namespace:

23.2.1 schema (Custom XML Schema Reference)

This element specifies the properties associated with a single XML namespace, for which all known XML schemas shall be loaded in order to validate the custom XML markup stored within this document. These properties can be used appropriately to locate custom XML schema(s) for use with the document. ISO/IEC 29500 does not require any particular XML schema language.

[*Note:* Some examples of XML schema languages that might be used to implement Custom XML Mappings include:

- W3C XML Schema - http://www.w3.org/XML/Schema
- RELAX NG – ISO/IEC 19757-2
- Schematron – ISO/IEC 19757-3
- NVDL – ISO/IEC 19757-4

end note]

[*Example*: Consider a WordprocessingML document which contains custom XML markup in the http://www.example.com namespace. The following content would be displayed in the document's schema library data:

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
<sl:schemaLibrary>
  <sl:schema sl:uri="http://www.example.com" sl:schemaLocation="c:\example.xsd"
/>
</sl:schemaLibrary>
```

The schema element contains the properties for this one XML namespace: in this case, a namespace URI of `http://www.example.com` and a file location of `c:\example.xsd`. *end example*]

[*Note*: This element is not intended to reintroduce transitional schema into the strict conformance class. *end note*]

Attributes	Description

manifestLocation (Supplementary XML File Location)	Specifies the location of a supplementary XML file that can be downloaded and parsed when this document is loaded in order to provide additional application-defined capabilities. The contents of this file are application-defined.
	
	[*Example*: Consider a WordprocessingML document that contains custom XML markup in the http://www.example.com namespace, which is associated with a resource file located at `http://www.example.com/resource.xml`. The following content would be displayed in the document's schema library data:
	
	`<sl:schemaLibrary>`
	`  <sl:schema sl:uri="http://www.example.com"`
	`sl:manifestLocation= "http://www.example.com/resource.xml" />`
	`</sl:schemaLibrary>`
	
	The manifestLocation attribute contains `http://www.example.com/resource.xml` which is the location of a resource file that can be downloaded for use when this namespace is used. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

schemaLanguage (Schema Language)	Specifies the media type or the root namespace of the schema language.
	
	[*Example*:
	
	`<sl:schema …`
	`schemaLanguage="http:/relaxng.org/ns/structure/1.0" />`
	
	end example]
	
	The possible values for this attribute are defined by the W3C XML Schema token datatype.

schemaLocation (Custom XML Schema Location)	Specifies the location of the XML schema file which should be downloaded and parsed when this document is loaded.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

Attributes	Description

	[*Example*: Consider a WordprocessingML document which contains custom XML markup in the http://www.example.com namespace, which is defined by an XML schema located at `c:\example.xsd`. The following content would be displayed in the document's schema library data:
	
	`<sl:schemaLibrary>`
	`  <sl:schema sl:uri="http://www.example.com" sl:schemaLocation=`
	`"c:\example.xsd" />`
	`</sl:schemaLibrary>`
	
	The schemaLocation attribute contains `c:\example.xsd` which is the location of the XML schema file used when this namespace is used. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

uri (Custom XML Schema Namespace)	Specifies the target namespace for the XML Schema associated with this schema reference.
	
	[*Example*: Consider the following content for custom XML namespace data:
	
	…
	`<sl:schema w:uri="http://www.example.com/schema1" />`
	`<sl:schema w:uri="http://www.example.com/schema2" />`
	…
	
	The uri attribute specifies the target namespace of each XML schema reference:
	• `http://www.example.com/schema1`
	• `http://www.example.com/schema2`
	
	Applications can then locate and utilize a schema for these namespaces using any means available. *end example*]
	
	The possible values for this attribute are defined by the W3C XML Schema string datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_Schema) is located in §A.7. *end note*]

23.2.2 schemaLibrary (Embedded Custom XML Schema Supplementary Data)

This element specifies the set of XML namespaces which have been associated with the contents of the custom XML markup within the current Office Open XML document. Each unique namespace which is referenced within the document can be referenced within this element by a single schema element, regardless of the number of constituent XML schemas which comprise that namespace.

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

[*Example*: Consider a WordprocessingML document which contains custom XML markup in two distinct namespaces: the http://www.example.com namespace and the http://www.example2.com namespace. If the first namespace is defined by a single XML schema, and the second is defined by five XML schemas (which are cross-referenced using the appropriate XML Schema syntax), the following content would be displayed in the document's schema library XML:

```
<sl:schemaLibrary>
  <sl:schema … />
  <sl:schema … />
</sl:schemaLibrary>
```

The schemaLibrary element contains only two schema elements even though there are six XML schemas in use, as there are only two distinct namespaces for which data is stored. *end example*]

[*Note*: The W3C XML Schema definition of this element's content model (CT_SchemaLibrary) is located in §A.7. *end note*]

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

Annex A.
(normative)
Schemas – W3C XML Schema

This Office Open XML specification includes a family of schemas defined using the W3C XML Schema 1.0 syntax. The normative definitions of these schemas follow below, and they also reside in an accompanying file named OfficeOpenXML-XMLSchema-Strict.zip, which is distributed in electronic form.

A.1 WordprocessingML

This schema is available in the file wml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:m="http://purl.oclc.org/ooxml/officeDocument/math"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:sl="http://purl.oclc.org/ooxml/schemaLibrary/main"
  xmlns:wp="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
  xmlns="http://purl.oclc.org/ooxml/wordprocessingml/main"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes" elementFormDefault="qualified"
  attributeFormDefault="qualified" blockDefault="#all"
  targetNamespace="http://purl.oclc.org/ooxml/wordprocessingml/main">
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/wordprocessingDrawing"
      schemaLocation="dml-wordprocessingDrawing.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/math" schemaLocation="shared-
      math.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/schemaLibrary/main " schemaLocation="shared-
      customXmlSchemaProperties.xsd"/>
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
    <xsd:complexType name="CT_Empty"/>
    <xsd:complexType name="CT_OnOff">
       <xsd:attribute name="val" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LongHexNumber">
       <xsd:restriction base="xsd:hexBinary">
          <xsd:length value="4"/>
       </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LongHexNumber">
       <xsd:attribute name="val" type="ST_LongHexNumber" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ShortHexNumber">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UcharHexNumber">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Charset">
        <xsd:attribute name="characterSet" type="s:ST_String" use="optional" default="ISO-8859-1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DecimalNumberOrPercent">
        <xsd:union memberTypes="s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DecimalNumber">
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_DecimalNumber">
        <xsd:attribute name="val" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_UnsignedDecimalNumber">
        <xsd:attribute name="val" type="s:ST_UnsignedDecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DecimalNumberOrPrecent">
        <xsd:attribute name="val" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TwipsMeasure">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SignedTwipsMeasure">
        <xsd:union memberTypes="xsd:integer s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_SignedTwipsMeasure">
        <xsd:attribute name="val" type="ST_SignedTwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PixelsMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_PixelsMeasure">
        <xsd:attribute name="val" type="ST_PixelsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HpsMeasure">
        <xsd:union memberTypes="s:ST_UnsignedDecimalNumber s:ST_PositiveUniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_HpsMeasure">
        <xsd:attribute name="val" type="ST_HpsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SignedHpsMeasure">
        <xsd:union memberTypes="xsd:integer s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_SignedHpsMeasure">
        <xsd:attribute name="val" type="ST_SignedHpsMeasure" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_DateTime">
        <xsd:restriction base="xsd:dateTime"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_MacroName">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="33"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MacroName">
        <xsd:attribute name="val" use="required" type="ST_MacroName"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EighthPointMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PointMeasure">
        <xsd:restriction base="s:ST_UnsignedDecimalNumber"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_String">
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextScale">
        <xsd:union memberTypes="ST_TextScalePercent"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextScalePercent">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="0*(600|([0-5]?[0-9]?[0-9]))%"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextScale">
        <xsd:attribute name="val" type="ST_TextScale"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HighlightColor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="blue"/>
            <xsd:enumeration value="cyan"/>
            <xsd:enumeration value="green"/>
            <xsd:enumeration value="magenta"/>
            <xsd:enumeration value="red"/>
            <xsd:enumeration value="yellow"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="darkBlue"/>
            <xsd:enumeration value="darkCyan"/>
            <xsd:enumeration value="darkGreen"/>
            <xsd:enumeration value="darkMagenta"/>
            <xsd:enumeration value="darkRed"/>
            <xsd:enumeration value="darkYellow"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_Highlight">
        <xsd:attribute name="val" type="ST_HighlightColor" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HexColorAuto">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HexColor">
        <xsd:union memberTypes="ST_HexColorAuto s:ST_HexColorRGB"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Color">
        <xsd:attribute name="val" type="ST_HexColor" use="required"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lang">
        <xsd:attribute name="val" type="s:ST_Lang" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Guid">
        <xsd:attribute name="val" type="s:ST_Guid"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Underline">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="words"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="dottedHeavy"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="dashedHeavy"/>
            <xsd:enumeration value="dashLong"/>
            <xsd:enumeration value="dashLongHeavy"/>
            <xsd:enumeration value="dotDash"/>
            <xsd:enumeration value="dashDotHeavy"/>
            <xsd:enumeration value="dotDotDash"/>
            <xsd:enumeration value="dashDotDotHeavy"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="wavyHeavy"/>
            <xsd:enumeration value="wavyDouble"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Underline">
        <xsd:attribute name="val" type="ST_Underline" use="optional"/>
        <xsd:attribute name="color" type="ST_HexColor" use="optional" default="auto"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextEffect">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="blinkBackground"/>
            <xsd:enumeration value="lights"/>
            <xsd:enumeration value="antsBlack"/>
            <xsd:enumeration value="antsRed"/>
            <xsd:enumeration value="shimmer"/>
            <xsd:enumeration value="sparkle"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextEffect">
        <xsd:attribute name="val" type="ST_TextEffect" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Border">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="dashed"/>
            <xsd:enumeration value="dotDash"/>
            <xsd:enumeration value="dotDotDash"/>
            <xsd:enumeration value="triple"/>
            <xsd:enumeration value="thinThickSmallGap"/>
            <xsd:enumeration value="thickThinSmallGap"/>
            <xsd:enumeration value="thinThickThinSmallGap"/>
            <xsd:enumeration value="thinThickMediumGap"/>
            <xsd:enumeration value="thickThinMediumGap"/>
            <xsd:enumeration value="thinThickThinMediumGap"/>
            <xsd:enumeration value="thinThickLargeGap"/>
            <xsd:enumeration value="thickThinLargeGap"/>
            <xsd:enumeration value="thinThickThinLargeGap"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="doubleWave"/>
            <xsd:enumeration value="dashSmallGap"/>
            <xsd:enumeration value="dashDotStroked"/>
            <xsd:enumeration value="threeDEmboss"/>
            <xsd:enumeration value="threeDEngrave"/>
            <xsd:enumeration value="outset"/>
            <xsd:enumeration value="inset"/>
            <xsd:enumeration value="apples"/>
            <xsd:enumeration value="archedScallops"/>
            <xsd:enumeration value="babyPacifier"/>
            <xsd:enumeration value="babyRattle"/>
            <xsd:enumeration value="balloons3Colors"/>
            <xsd:enumeration value="balloonsHotAir"/>
            <xsd:enumeration value="basicBlackDashes"/>
            <xsd:enumeration value="basicBlackDots"/>
            <xsd:enumeration value="basicBlackSquares"/>
            <xsd:enumeration value="basicThinLines"/>
            <xsd:enumeration value="basicWhiteDashes"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="basicWhiteDots"/>
            <xsd:enumeration value="basicWhiteSquares"/>
            <xsd:enumeration value="basicWideInline"/>
            <xsd:enumeration value="basicWideMidline"/>
            <xsd:enumeration value="basicWideOutline"/>
            <xsd:enumeration value="bats"/>
            <xsd:enumeration value="birds"/>
            <xsd:enumeration value="birdsFlight"/>
            <xsd:enumeration value="cabins"/>
            <xsd:enumeration value="cakeSlice"/>
            <xsd:enumeration value="candyCorn"/>
            <xsd:enumeration value="celticKnotwork"/>
            <xsd:enumeration value="certificateBanner"/>
            <xsd:enumeration value="chainLink"/>
            <xsd:enumeration value="champagneBottle"/>
            <xsd:enumeration value="checkedBarBlack"/>
            <xsd:enumeration value="checkedBarColor"/>
            <xsd:enumeration value="checkered"/>
            <xsd:enumeration value="christmasTree"/>
            <xsd:enumeration value="circlesLines"/>
            <xsd:enumeration value="circlesRectangles"/>
            <xsd:enumeration value="classicalWave"/>
            <xsd:enumeration value="clocks"/>
            <xsd:enumeration value="compass"/>
            <xsd:enumeration value="confetti"/>
            <xsd:enumeration value="confettiGrays"/>
            <xsd:enumeration value="confettiOutline"/>
            <xsd:enumeration value="confettiStreamers"/>
            <xsd:enumeration value="confettiWhite"/>
            <xsd:enumeration value="cornerTriangles"/>
            <xsd:enumeration value="couponCutoutDashes"/>
            <xsd:enumeration value="couponCutoutDots"/>
            <xsd:enumeration value="crazyMaze"/>
            <xsd:enumeration value="creaturesButterfly"/>
            <xsd:enumeration value="creaturesFish"/>
            <xsd:enumeration value="creaturesInsects"/>
            <xsd:enumeration value="creaturesLadyBug"/>
            <xsd:enumeration value="crossStitch"/>
            <xsd:enumeration value="cup"/>
            <xsd:enumeration value="decoArch"/>
            <xsd:enumeration value="decoArchColor"/>
            <xsd:enumeration value="decoBlocks"/>
            <xsd:enumeration value="diamondsGray"/>
            <xsd:enumeration value="doubleD"/>
            <xsd:enumeration value="doubleDiamonds"/>
            <xsd:enumeration value="earth1"/>
            <xsd:enumeration value="earth2"/>
            <xsd:enumeration value="earth3"/>
            <xsd:enumeration value="eclipsingSquares1"/>
            <xsd:enumeration value="eclipsingSquares2"/>
            <xsd:enumeration value="eggsBlack"/>
            <xsd:enumeration value="fans"/>
            <xsd:enumeration value="film"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="firecrackers"/>
            <xsd:enumeration value="flowersBlockPrint"/>
            <xsd:enumeration value="flowersDaisies"/>
            <xsd:enumeration value="flowersModern1"/>
            <xsd:enumeration value="flowersModern2"/>
            <xsd:enumeration value="flowersPansy"/>
            <xsd:enumeration value="flowersRedRose"/>
            <xsd:enumeration value="flowersRoses"/>
            <xsd:enumeration value="flowersTeacup"/>
            <xsd:enumeration value="flowersTiny"/>
            <xsd:enumeration value="gems"/>
            <xsd:enumeration value="gingerbreadMan"/>
            <xsd:enumeration value="gradient"/>
            <xsd:enumeration value="handmade1"/>
            <xsd:enumeration value="handmade2"/>
            <xsd:enumeration value="heartBalloon"/>
            <xsd:enumeration value="heartGray"/>
            <xsd:enumeration value="hearts"/>
            <xsd:enumeration value="heebieJeebies"/>
            <xsd:enumeration value="holly"/>
            <xsd:enumeration value="houseFunky"/>
            <xsd:enumeration value="hypnotic"/>
            <xsd:enumeration value="iceCreamCones"/>
            <xsd:enumeration value="lightBulb"/>
            <xsd:enumeration value="lightning1"/>
            <xsd:enumeration value="lightning2"/>
            <xsd:enumeration value="mapPins"/>
            <xsd:enumeration value="mapleLeaf"/>
            <xsd:enumeration value="mapleMuffins"/>
            <xsd:enumeration value="marquee"/>
            <xsd:enumeration value="marqueeToothed"/>
            <xsd:enumeration value="moons"/>
            <xsd:enumeration value="mosaic"/>
            <xsd:enumeration value="musicNotes"/>
            <xsd:enumeration value="northwest"/>
            <xsd:enumeration value="ovals"/>
            <xsd:enumeration value="packages"/>
            <xsd:enumeration value="palmsBlack"/>
            <xsd:enumeration value="palmsColor"/>
            <xsd:enumeration value="paperClips"/>
            <xsd:enumeration value="papyrus"/>
            <xsd:enumeration value="partyFavor"/>
            <xsd:enumeration value="partyGlass"/>
            <xsd:enumeration value="pencils"/>
            <xsd:enumeration value="people"/>
            <xsd:enumeration value="peopleWaving"/>
            <xsd:enumeration value="peopleHats"/>
            <xsd:enumeration value="poinsettias"/>
            <xsd:enumeration value="postageStamp"/>
            <xsd:enumeration value="pumpkin1"/>
            <xsd:enumeration value="pushPinNote2"/>
            <xsd:enumeration value="pushPinNote1"/>
            <xsd:enumeration value="pyramids"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="pyramidsAbove"/>
            <xsd:enumeration value="quadrants"/>
            <xsd:enumeration value="rings"/>
            <xsd:enumeration value="safari"/>
            <xsd:enumeration value="sawtooth"/>
            <xsd:enumeration value="sawtoothGray"/>
            <xsd:enumeration value="scaredCat"/>
            <xsd:enumeration value="seattle"/>
            <xsd:enumeration value="shadowedSquares"/>
            <xsd:enumeration value="sharksTeeth"/>
            <xsd:enumeration value="shorebirdTracks"/>
            <xsd:enumeration value="skyrocket"/>
            <xsd:enumeration value="snowflakeFancy"/>
            <xsd:enumeration value="snowflakes"/>
            <xsd:enumeration value="sombrero"/>
            <xsd:enumeration value="southwest"/>
            <xsd:enumeration value="stars"/>
            <xsd:enumeration value="starsTop"/>
            <xsd:enumeration value="stars3d"/>
            <xsd:enumeration value="starsBlack"/>
            <xsd:enumeration value="starsShadowed"/>
            <xsd:enumeration value="sun"/>
            <xsd:enumeration value="swirligig"/>
            <xsd:enumeration value="tornPaper"/>
            <xsd:enumeration value="tornPaperBlack"/>
            <xsd:enumeration value="trees"/>
            <xsd:enumeration value="triangleParty"/>
            <xsd:enumeration value="triangles"/>
            <xsd:enumeration value="triangle1"/>
            <xsd:enumeration value="triangle2"/>
            <xsd:enumeration value="triangleCircle1"/>
            <xsd:enumeration value="triangleCircle2"/>
            <xsd:enumeration value="shapes1"/>
            <xsd:enumeration value="shapes2"/>
            <xsd:enumeration value="twistedLines1"/>
            <xsd:enumeration value="twistedLines2"/>
            <xsd:enumeration value="vine"/>
            <xsd:enumeration value="waveline"/>
            <xsd:enumeration value="weavingAngles"/>
            <xsd:enumeration value="weavingBraid"/>
            <xsd:enumeration value="weavingRibbon"/>
            <xsd:enumeration value="weavingStrips"/>
            <xsd:enumeration value="whiteFlowers"/>
            <xsd:enumeration value="woodwork"/>
            <xsd:enumeration value="xIllusions"/>
            <xsd:enumeration value="zanyTriangles"/>
            <xsd:enumeration value="zigZag"/>
            <xsd:enumeration value="zigZagStitch"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Border">
        <xsd:attribute name="val" type="ST_Border" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="color" type="ST_HexColor" use="optional" default="auto"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="sz" type="ST_EighthPointMeasure" use="optional"/>
        <xsd:attribute name="space" type="ST_PointMeasure" use="optional" default="0"/>
        <xsd:attribute name="shadow" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="frame" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Shd">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="horzStripe"/>
            <xsd:enumeration value="vertStripe"/>
            <xsd:enumeration value="reverseDiagStripe"/>
            <xsd:enumeration value="diagStripe"/>
            <xsd:enumeration value="horzCross"/>
            <xsd:enumeration value="diagCross"/>
            <xsd:enumeration value="thinHorzStripe"/>
            <xsd:enumeration value="thinVertStripe"/>
            <xsd:enumeration value="thinReverseDiagStripe"/>
            <xsd:enumeration value="thinDiagStripe"/>
            <xsd:enumeration value="thinHorzCross"/>
            <xsd:enumeration value="thinDiagCross"/>
            <xsd:enumeration value="pct5"/>
            <xsd:enumeration value="pct10"/>
            <xsd:enumeration value="pct12"/>
            <xsd:enumeration value="pct15"/>
            <xsd:enumeration value="pct20"/>
            <xsd:enumeration value="pct25"/>
            <xsd:enumeration value="pct30"/>
            <xsd:enumeration value="pct35"/>
            <xsd:enumeration value="pct37"/>
            <xsd:enumeration value="pct40"/>
            <xsd:enumeration value="pct45"/>
            <xsd:enumeration value="pct50"/>
            <xsd:enumeration value="pct55"/>
            <xsd:enumeration value="pct60"/>
            <xsd:enumeration value="pct62"/>
            <xsd:enumeration value="pct65"/>
            <xsd:enumeration value="pct70"/>
            <xsd:enumeration value="pct75"/>
            <xsd:enumeration value="pct80"/>
            <xsd:enumeration value="pct85"/>
            <xsd:enumeration value="pct87"/>
            <xsd:enumeration value="pct90"/>
            <xsd:enumeration value="pct95"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shd">
        <xsd:attribute name="val" type="ST_Shd" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="color" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="fill" type="ST_HexColor" use="optional"/>
        <xsd:attribute name="themeFill" type="ST_ThemeColor" use="optional"/>
        <xsd:attribute name="themeFillTint" type="ST_UcharHexNumber" use="optional"/>
        <xsd:attribute name="themeFillShade" type="ST_UcharHexNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VerticalAlignRun">
        <xsd:attribute name="val" type="s:ST_VerticalAlignRun" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FitText">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Em">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="comma"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="underDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Em">
        <xsd:attribute name="val" type="ST_Em" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Language">
        <xsd:attribute name="val" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="eastAsia" type="s:ST_Lang" use="optional"/>
        <xsd:attribute name="bidi" type="s:ST_Lang" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CombineBrackets">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="round"/>
            <xsd:enumeration value="square"/>
            <xsd:enumeration value="angle"/>
            <xsd:enumeration value="curly"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_EastAsianLayout">
        <xsd:attribute name="id" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="combine" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="combineBrackets" type="ST_CombineBrackets" use="optional"/>
        <xsd:attribute name="vert" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="vertCompress" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HeightRule">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="exact"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="atLeast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Wrap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="notBeside"/>
            <xsd:enumeration value="around"/>
            <xsd:enumeration value="tight"/>
            <xsd:enumeration value="through"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HAnchor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="page"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DropCap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="drop"/>
            <xsd:enumeration value="margin"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FramePr">
        <xsd:attribute name="dropCap" type="ST_DropCap" use="optional"/>
        <xsd:attribute name="lines" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="w" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="h" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="vSpace" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="hSpace" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="wrap" type="ST_Wrap" use="optional"/>
        <xsd:attribute name="hAnchor" type="ST_HAnchor" use="optional"/>
        <xsd:attribute name="vAnchor" type="ST_VAnchor" use="optional"/>
        <xsd:attribute name="x" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="xAlign" type="s:ST_XAlign" use="optional"/>
        <xsd:attribute name="y" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="yAlign" type="s:ST_YAlign" use="optional"/>
        <xsd:attribute name="hRule" type="ST_HeightRule" use="optional"/>
        <xsd:attribute name="anchorLock" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TabJc">
        <xsd:restriction base="xsd:string">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="start"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="bar"/>
            <xsd:enumeration value="num"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TabTlc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="underscore"/>
            <xsd:enumeration value="heavy"/>
            <xsd:enumeration value="middleDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TabStop">
        <xsd:attribute name="val" type="ST_TabJc" use="required"/>
        <xsd:attribute name="leader" type="ST_TabTlc" use="optional"/>
        <xsd:attribute name="pos" type="ST_SignedTwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LineSpacingRule">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="exact"/>
            <xsd:enumeration value="atLeast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Spacing">
        <xsd:attribute name="before" type="s:ST_TwipsMeasure" use="optional" default="0"/>
        <xsd:attribute name="beforeLines" type="ST_DecimalNumber" use="optional" default="0"/>
        <xsd:attribute name="beforeAutospacing" type="s:ST_OnOff" use="optional" default="off"/>
        <xsd:attribute name="after" type="s:ST_TwipsMeasure" use="optional" default="0"/>
        <xsd:attribute name="afterLines" type="ST_DecimalNumber" use="optional" default="0"/>
        <xsd:attribute name="afterAutospacing" type="s:ST_OnOff" use="optional" default="off"/>
        <xsd:attribute name="line" type="ST_SignedTwipsMeasure" use="optional" default="0"/>
        <xsd:attribute name="lineRule" type="ST_LineSpacingRule" use="optional" default="auto"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ind">
        <xsd:attribute name="start" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="startChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="end" type="ST_SignedTwipsMeasure" use="optional"/>
        <xsd:attribute name="endChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="hanging" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="hangingChars" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="firstLine" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="firstLineChars" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Jc">
        <xsd:restriction base="xsd:string">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="start"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="mediumKashida"/>
            <xsd:enumeration value="distribute"/>
            <xsd:enumeration value="numTab"/>
            <xsd:enumeration value="highKashida"/>
            <xsd:enumeration value="lowKashida"/>
            <xsd:enumeration value="thaiDistribute"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_JcTable">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="start"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Jc">
        <xsd:attribute name="val" type="ST_Jc" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_JcTable">
        <xsd:attribute name="val" type="ST_JcTable" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_View">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="print"/>
            <xsd:enumeration value="outline"/>
            <xsd:enumeration value="masterPages"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="web"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_View">
        <xsd:attribute name="val" type="ST_View" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Zoom">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="fullPage"/>
            <xsd:enumeration value="bestFit"/>
            <xsd:enumeration value="textFit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Zoom">
        <xsd:attribute name="val" type="ST_Zoom" use="optional"/>
        <xsd:attribute name="percent" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WritingStyle">
        <xsd:attribute name="lang" type="s:ST_Lang" use="required"/>
        <xsd:attribute name="vendorID" type="s:ST_String" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="dllVersion" type="s:ST_String" use="required"/>
        <xsd:attribute name="nlCheck" type="s:ST_OnOff" use="optional" default="off"/>
        <xsd:attribute name="checkStyle" type="s:ST_OnOff" use="required"/>
        <xsd:attribute name="appName" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Proof">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="clean"/>
            <xsd:enumeration value="dirty"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Proof">
        <xsd:attribute name="spelling" type="ST_Proof" use="optional"/>
        <xsd:attribute name="grammar" type="ST_Proof" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocType">
        <xsd:attribute name="val" type="ST_DocType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocProtect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="readOnly"/>
            <xsd:enumeration value="comments"/>
            <xsd:enumeration value="trackedChanges"/>
            <xsd:enumeration value="forms"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Password">
        <xsd:attribute name="algorithmName" type="s:ST_String" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="ST_DecimalNumber" use="optional"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_DocProtect">
        <xsd:attribute name="edit" type="ST_DocProtect" use="optional"/>
        <xsd:attribute name="formatting" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="enforcement" type="s:ST_OnOff"/>
        <xsd:attributeGroup ref="AG_Password"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDocType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="catalog"/>
            <xsd:enumeration value="envelopes"/>
            <xsd:enumeration value="mailingLabels"/>
            <xsd:enumeration value="formLetters"/>
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="fax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDocType">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="val" type="ST_MailMergeDocType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDataType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDataType">
        <xsd:attribute name="val" type="ST_MailMergeDataType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeDest">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="newDocument"/>
            <xsd:enumeration value="printer"/>
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="fax"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeDest">
        <xsd:attribute name="val" type="ST_MailMergeDest" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeOdsoFMDFieldType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="null"/>
            <xsd:enumeration value="dbColumn"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeOdsoFMDFieldType">
        <xsd:attribute name="val" type="ST_MailMergeOdsoFMDFieldType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChangesView">
        <xsd:attribute name="markup" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="comments" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="insDel" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="formatting" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="inkAnnotations" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Kinsoku">
        <xsd:attribute name="lang" type="s:ST_Lang" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextDirection">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="tb"/>
            <xsd:enumeration value="rl"/>
            <xsd:enumeration value="lr"/>
            <xsd:enumeration value="tbV"/>
            <xsd:enumeration value="rlV"/>
            <xsd:enumeration value="lrV"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextDirection">
        <xsd:attribute name="val" type="ST_TextDirection" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextAlignment">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="baseline"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextAlignment">
        <xsd:attribute name="val" type="ST_TextAlignment" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DisplacedByCustomXml">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="next"/>
            <xsd:enumeration value="prev"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnnotationVMerge">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cont"/>
            <xsd:enumeration value="rest"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Markup">
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
                <xsd:attribute name="author" type="s:ST_String" use="required"/>
                <xsd:attribute name="date" type="ST_DateTime" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_CellMergeTrackChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:attribute name="vMerge" type="ST_AnnotationVMerge" use="optional"/>
                <xsd:attribute name="vMergeOrig" type="ST_AnnotationVMerge" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TrackChangeRange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml"
                  use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_MarkupRange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml"
              use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_BookmarkRange">
    <xsd:complexContent>
        <xsd:extension base="CT_MarkupRange">
            <xsd:attribute name="colFirst" type="ST_DecimalNumber" use="optional"/>
            <xsd:attribute name="colLast" type="ST_DecimalNumber" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Bookmark">
    <xsd:complexContent>
        <xsd:extension base="CT_BookmarkRange">
            <xsd:attribute name="name" type="s:ST_String" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_MoveBookmark">
    <xsd:complexContent>
        <xsd:extension base="CT_Bookmark">
            <xsd:attribute name="author" type="s:ST_String" use="required"/>
            <xsd:attribute name="date" type="ST_DateTime" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Comment">
    <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
            <xsd:sequence>
                <xsd:group ref="EG_BlockLevelElts" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="initials" type="s:ST_String" use="optional"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_TblPrExChange">
    <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
            <xsd:sequence>
                <xsd:element name="tblPrEx" type="CT_TblPrExBase" minOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_TcPrChange">
    <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
            <xsd:sequence>
                <xsd:element name="tcPr" type="CT_TcPrInner" minOccurs="1"/>
            </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="trPr" type="CT_TrPrBase" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridChange">
        <xsd:complexContent>
            <xsd:extension base="CT_Markup">
                <xsd:sequence>
                    <xsd:element name="tblGrid" type="CT_TblGridBase"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="tblPr" type="CT_TblPrBase"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_SectPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="sectPr" type="CT_SectPrBase" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="pPr" type="CT_PPrBase" minOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrChange">
        <xsd:complexContent>
            <xsd:extension base="CT_TrackChange">
                <xsd:sequence>
                    <xsd:element name="rPr" type="CT_RPrOriginal" minOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_ParaRPrChange">
    <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
            <xsd:sequence>
                <xsd:element name="rPr" type="CT_ParaRPrOriginal" minOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_RunTrackChange">
    <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:group ref="EG_ContentRunContent"/>
                <xsd:group ref="m:EG_OMathMathElements"/>
            </xsd:choice>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:group name="EG_PContentMath">
  <xsd:choice>
    <xsd:group ref="EG_PContentBase" minOccurs="0" maxOccurs="unbounded" />
    <xsd:group ref="EG_ContentRunContentBase" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:choice>
</xsd:group>
<xsd:group name="EG_PContentBase">
  <xsd:choice>
    <xsd:element name="customXml" type="CT_CustomXmlRun"/>
    <xsd:element name="fldSimple" type="CT_SimpleField" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element name="hyperlink" type="CT_Hyperlink"/>
  </xsd:choice>
</xsd:group>
<xsd:group name="EG_ContentRunContentBase">
  <xsd:choice>
    <xsd:element name="smartTag" type="CT_SmartTagRun"/>
    <xsd:element name="sdt" type="CT_SdtRun"/>
    <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded" />
  </xsd:choice>
</xsd:group>
<xsd:group name="EG_CellMarkupElements">
    <xsd:choice>
        <xsd:element name="cellIns" type="CT_TrackChange" minOccurs="0"/>
        <xsd:element name="cellDel" type="CT_TrackChange" minOccurs="0"/>
        <xsd:element name="cellMerge" type="CT_CellMergeTrackChange" minOccurs="0"/>
    </xsd:choice>
</xsd:group>
<xsd:group name="EG_RangeMarkupElements">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:choice>
            <xsd:element name="bookmarkStart" type="CT_Bookmark"/>
            <xsd:element name="bookmarkEnd" type="CT_MarkupRange"/>
            <xsd:element name="moveFromRangeStart" type="CT_MoveBookmark"/>
            <xsd:element name="moveFromRangeEnd" type="CT_MarkupRange"/>
            <xsd:element name="moveToRangeStart" type="CT_MoveBookmark"/>
            <xsd:element name="moveToRangeEnd" type="CT_MarkupRange"/>
            <xsd:element name="commentRangeStart" type="CT_MarkupRange"/>
            <xsd:element name="commentRangeEnd" type="CT_MarkupRange"/>
            <xsd:element name="customXmlInsRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlInsRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlDelRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlDelRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlMoveFromRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlMoveFromRangeEnd" type="CT_Markup"/>
            <xsd:element name="customXmlMoveToRangeStart" type="CT_TrackChange"/>
            <xsd:element name="customXmlMoveToRangeEnd" type="CT_Markup"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_NumPr">
        <xsd:sequence>
            <xsd:element name="ilvl" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PBdr">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
            <xsd:element name="between" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bar" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Tabs">
        <xsd:sequence>
            <xsd:element name="tab" type="CT_TabStop" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextboxTightWrap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="allLines"/>
            <xsd:enumeration value="firstAndLastLine"/>
            <xsd:enumeration value="firstLineOnly"/>
            <xsd:enumeration value="lastLineOnly"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextboxTightWrap">
        <xsd:attribute name="val" type="ST_TextboxTightWrap" use="required"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_PPr">
        <xsd:complexContent>
            <xsd:extension base="CT_PPrBase">
                <xsd:sequence>
                    <xsd:element name="rPr" type="CT_ParaRPr" minOccurs="0"/>
                    <xsd:element name="sectPr" type="CT_SectPr" minOccurs="0"/>
                    <xsd:element name="pPrChange" type="CT_PPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrBase">
        <xsd:sequence>
            <xsd:element name="pStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="keepNext" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="keepLines" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pageBreakBefore" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="framePr" type="CT_FramePr" minOccurs="0"/>
            <xsd:element name="widowControl" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="numPr" type="CT_NumPr" minOccurs="0"/>
            <xsd:element name="suppressLineNumbers" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pBdr" type="CT_PBdr" minOccurs="0"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0"/>
            <xsd:element name="tabs" type="CT_Tabs" minOccurs="0"/>
            <xsd:element name="suppressAutoHyphens" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="kinsoku" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="wordWrap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="overflowPunct" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="topLinePunct" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autoSpaceDE" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="autoSpaceDN" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bidi" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="adjustRightInd" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="snapToGrid" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="spacing" type="CT_Spacing" minOccurs="0"/>
            <xsd:element name="ind" type="CT_Ind" minOccurs="0"/>
            <xsd:element name="contextualSpacing" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="mirrorIndents" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suppressOverlap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="jc" type="CT_Jc" minOccurs="0"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0"/>
            <xsd:element name="textAlignment" type="CT_TextAlignment" minOccurs="0"/>
            <xsd:element name="textboxTightWrap" type="CT_TextboxTightWrap" minOccurs="0"/>
            <xsd:element name="outlineLvl" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="divId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrGeneral">
        <xsd:complexContent>
            <xsd:extension base="CT_PPrBase">
                <xsd:sequence>
                    <xsd:element name="pPrChange" type="CT_PPrChange" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CT_Control">
    <xsd:attribute name="name" type="s:ST_String" use="optional"/>
    <xsd:attribute name="shapeid" type="s:ST_String" use="optional"/>
    <xsd:attribute ref="r:id" use="optional"/>
</xsd:complexType>
<xsd:complexType name="CT_Background">
    <xsd:sequence>
        <xsd:element name="drawing" type="CT_Drawing" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="color" type="ST_HexColor" use="optional" default="auto"/>
    <xsd:attribute name="themeColor" type="ST_ThemeColor" use="optional"/>
    <xsd:attribute name="themeTint" type="ST_UcharHexNumber" use="optional"/>
    <xsd:attribute name="themeShade" type="ST_UcharHexNumber" use="optional"/>
</xsd:complexType>
<xsd:complexType name="CT_Rel">
    <xsd:attribute ref="r:id" use="required"/>
</xsd:complexType>
<xsd:complexType name="CT_Object">
    <xsd:sequence>
        <xsd:element name="drawing" type="CT_Drawing" minOccurs="0"/>
        <xsd:choice minOccurs="0">
            <xsd:element name="control" type="CT_Control"/>
            <xsd:element name="objectLink" type="CT_ObjectLink"/>
            <xsd:element name="objectEmbed" type="CT_ObjectEmbed"/>
            <xsd:element name="movie" type="CT_Rel"/>
        </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="dxaOrig" type="s:ST_TwipsMeasure" use="optional"/>
    <xsd:attribute name="dyaOrig" type="s:ST_TwipsMeasure" use="optional"/>
</xsd:complexType>
<xsd:complexType name="CT_ObjectEmbed">
    <xsd:attribute name="drawAspect" type="ST_ObjectDrawAspect" use="optional"/>
    <xsd:attribute ref="r:id" use="required"/>
    <xsd:attribute name="progId" type="s:ST_String" use="optional"/>
    <xsd:attribute name="shapeId" type="s:ST_String" use="optional"/>
    <xsd:attribute name="fieldCodes" type="s:ST_String" use="optional"/>
</xsd:complexType>
<xsd:simpleType name="ST_ObjectDrawAspect">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="content"/>
        <xsd:enumeration value="icon"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CT_ObjectLink">
    <xsd:complexContent>
        <xsd:extension base="CT_ObjectEmbed">
            <xsd:attribute name="updateMode" type="ST_ObjectUpdateMode" use="required"/>
            <xsd:attribute name="lockedField" type="s:ST_OnOff" use="optional"/>
        </xsd:extension>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_ObjectUpdateMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="always"/>
            <xsd:enumeration value="onCall"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Drawing">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element ref="wp:anchor" minOccurs="0"/>
            <xsd:element ref="wp:inline" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_SimpleField">
        <xsd:sequence>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="instr" type="s:ST_String" use="required"/>
        <xsd:attribute name="fldLock" type="s:ST_OnOff"/>
        <xsd:attribute name="dirty" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FldCharType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="begin"/>
            <xsd:enumeration value="separate"/>
            <xsd:enumeration value="end"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_InfoTextType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="autoText"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFHelpTextVal">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="256"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFStatusTextVal">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="140"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFName">
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="65"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FFTextType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="regular"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="number"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="currentTime"/>
            <xsd:enumeration value="currentDate"/>
            <xsd:enumeration value="calculated"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FFTextType">
        <xsd:attribute name="val" type="ST_FFTextType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFName">
        <xsd:attribute name="val" type="ST_FFName"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FldChar">
        <xsd:choice>
            <xsd:element name="ffData" type="CT_FFData" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="fldCharType" type="ST_FldCharType" use="required"/>
        <xsd:attribute name="fldLock" type="s:ST_OnOff"/>
        <xsd:attribute name="dirty" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="tgtFrame" type="s:ST_String" use="optional"/>
        <xsd:attribute name="tooltip" type="s:ST_String" use="optional"/>
        <xsd:attribute name="docLocation" type="s:ST_String" use="optional"/>
        <xsd:attribute name="history" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="anchor" type="s:ST_String" use="optional"/>
        <xsd:attribute ref="r:id"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFData">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="name" type="CT_FFName"/>
            <xsd:element name="label" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="tabIndex" type="CT_UnsignedDecimalNumber" minOccurs="0"/>
            <xsd:element name="enabled" type="CT_OnOff"/>
            <xsd:element name="calcOnExit" type="CT_OnOff"/>
            <xsd:element name="entryMacro" type="CT_MacroName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="exitMacro" type="CT_MacroName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="helpText" type="CT_FFHelpText" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="statusText" type="CT_FFStatusText" minOccurs="0" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="checkBox" type="CT_FFCheckBox"/>
                <xsd:element name="ddList" type="CT_FFDDList"/>
                <xsd:element name="textInput" type="CT_FFTextInput"/>
            </xsd:choice>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_FFHelpText">
        <xsd:attribute name="type" type="ST_InfoTextType"/>
        <xsd:attribute name="val" type="ST_FFHelpTextVal"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFStatusText">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="type" type="ST_InfoTextType"/>
        <xsd:attribute name="val" type="ST_FFStatusTextVal"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FFCheckBox">
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="size" type="CT_HpsMeasure"/>
                <xsd:element name="sizeAuto" type="CT_OnOff"/>
            </xsd:choice>
            <xsd:element name="default" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="checked" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FFDDList">
        <xsd:sequence>
            <xsd:element name="result" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="default" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="listEntry" type="CT_String" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FFTextInput">
        <xsd:sequence>
            <xsd:element name="type" type="CT_FFTextType" minOccurs="0"/>
            <xsd:element name="default" type="CT_String" minOccurs="0"/>
            <xsd:element name="maxLength" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="format" type="CT_String" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SectionMark">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nextPage"/>
            <xsd:enumeration value="nextColumn"/>
            <xsd:enumeration value="continuous"/>
            <xsd:enumeration value="evenPage"/>
            <xsd:enumeration value="oddPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SectType">
        <xsd:attribute name="val" type="ST_SectionMark"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PaperSource">
        <xsd:attribute name="first" type="ST_DecimalNumber"/>
        <xsd:attribute name="other" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_NumberFormat">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="upperRoman"/>
            <xsd:enumeration value="lowerRoman"/>
            <xsd:enumeration value="upperLetter"/>
            <xsd:enumeration value="lowerLetter"/>
            <xsd:enumeration value="ordinal"/>
            <xsd:enumeration value="cardinalText"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="ordinalText"/>
            <xsd:enumeration value="hex"/>
            <xsd:enumeration value="chicago"/>
            <xsd:enumeration value="ideographDigital"/>
            <xsd:enumeration value="japaneseCounting"/>
            <xsd:enumeration value="aiueo"/>
            <xsd:enumeration value="iroha"/>
            <xsd:enumeration value="decimalFullWidth"/>
            <xsd:enumeration value="decimalHalfWidth"/>
            <xsd:enumeration value="japaneseLegal"/>
            <xsd:enumeration value="japaneseDigitalTenThousand"/>
            <xsd:enumeration value="decimalEnclosedCircle"/>
            <xsd:enumeration value="decimalFullWidth2"/>
            <xsd:enumeration value="aiueoFullWidth"/>
            <xsd:enumeration value="irohaFullWidth"/>
            <xsd:enumeration value="decimalZero"/>
            <xsd:enumeration value="bullet"/>
            <xsd:enumeration value="ganada"/>
            <xsd:enumeration value="chosung"/>
            <xsd:enumeration value="decimalEnclosedFullstop"/>
            <xsd:enumeration value="decimalEnclosedParen"/>
            <xsd:enumeration value="decimalEnclosedCircleChinese"/>
            <xsd:enumeration value="ideographEnclosedCircle"/>
            <xsd:enumeration value="ideographTraditional"/>
            <xsd:enumeration value="ideographZodiac"/>
            <xsd:enumeration value="ideographZodiacTraditional"/>
            <xsd:enumeration value="taiwaneseCounting"/>
            <xsd:enumeration value="ideographLegalTraditional"/>
            <xsd:enumeration value="taiwaneseCountingThousand"/>
            <xsd:enumeration value="taiwaneseDigital"/>
            <xsd:enumeration value="chineseCounting"/>
            <xsd:enumeration value="chineseLegalSimplified"/>
            <xsd:enumeration value="chineseCountingThousand"/>
            <xsd:enumeration value="koreanDigital"/>
            <xsd:enumeration value="koreanCounting"/>
            <xsd:enumeration value="koreanLegal"/>
            <xsd:enumeration value="koreanDigital2"/>
            <xsd:enumeration value="vietnameseCounting"/>
            <xsd:enumeration value="russianLower"/>
            <xsd:enumeration value="russianUpper"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="numberInDash"/>
            <xsd:enumeration value="hebrew1"/>
            <xsd:enumeration value="hebrew2"/>
            <xsd:enumeration value="arabicAlpha"/>
            <xsd:enumeration value="arabicAbjad"/>
            <xsd:enumeration value="hindiVowels"/>
            <xsd:enumeration value="hindiConsonants"/>
            <xsd:enumeration value="hindiNumbers"/>
            <xsd:enumeration value="hindiCounting"/>
            <xsd:enumeration value="thaiLetters"/>
            <xsd:enumeration value="thaiNumbers"/>
            <xsd:enumeration value="thaiCounting"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="bahtText"/>
            <xsd:enumeration value="dollarText"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageOrientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="portrait"/>
            <xsd:enumeration value="landscape"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageSz">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="h" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="orient" type="ST_PageOrientation" use="optional"/>
        <xsd:attribute name="code" type="ST_DecimalNumber" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageMar">
        <xsd:attribute name="top" type="ST_SignedTwipsMeasure" use="required"/>
        <xsd:attribute name="right" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="bottom" type="ST_SignedTwipsMeasure" use="required"/>
        <xsd:attribute name="left" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="header" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="footer" type="s:ST_TwipsMeasure" use="required"/>
        <xsd:attribute name="gutter" type="s:ST_TwipsMeasure" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PageBorderZOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="front"/>
            <xsd:enumeration value="back"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageBorderDisplay">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="allPages"/>
            <xsd:enumeration value="firstPage"/>
            <xsd:enumeration value="notFirstPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PageBorderOffset">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="text"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageBorders">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TopPageBorder" minOccurs="0"/>
            <xsd:element name="left" type="CT_PageBorder" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_BottomPageBorder" minOccurs="0"/>
            <xsd:element name="right" type="CT_PageBorder" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="zOrder" type="ST_PageBorderZOrder" use="optional" default="front"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="display" type="ST_PageBorderDisplay" use="optional"/>
        <xsd:attribute name="offsetFrom" type="ST_PageBorderOffset" use="optional" default="text"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_Border">
                <xsd:attribute ref="r:id" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_BottomPageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_PageBorder">
                <xsd:attribute ref="r:bottomLeft" use="optional"/>
                <xsd:attribute ref="r:bottomRight" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TopPageBorder">
        <xsd:complexContent>
            <xsd:extension base="CT_PageBorder">
                <xsd:attribute ref="r:topLeft" use="optional"/>
                <xsd:attribute ref="r:topRight" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_ChapterSep">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="period"/>
            <xsd:enumeration value="colon"/>
            <xsd:enumeration value="emDash"/>
            <xsd:enumeration value="enDash"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LineNumberRestart">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="newPage"/>
            <xsd:enumeration value="newSection"/>
            <xsd:enumeration value="continuous"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LineNumber">
        <xsd:attribute name="countBy" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="start" type="ST_DecimalNumber" use="optional" default="1"/>
        <xsd:attribute name="distance" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="restart" type="ST_LineNumberRestart" use="optional" default="newPage"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageNumber">
        <xsd:attribute name="fmt" type="ST_NumberFormat" use="optional" default="decimal"/>
        <xsd:attribute name="start" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="chapStyle" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="chapSep" type="ST_ChapterSep" use="optional" default="hyphen"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_Column">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure" use="optional"/>
        <xsd:attribute name="space" type="s:ST_TwipsMeasure" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Columns">
        <xsd:sequence minOccurs="0">
            <xsd:element name="col" type="CT_Column" maxOccurs="45"/>
        </xsd:sequence>
        <xsd:attribute name="equalWidth" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="space" type="s:ST_TwipsMeasure" use="optional" default="720"/>
        <xsd:attribute name="num" type="ST_DecimalNumber" use="optional" default="1"/>
        <xsd:attribute name="sep" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_VerticalJc">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="both"/>
            <xsd:enumeration value="bottom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_VerticalJc">
        <xsd:attribute name="val" type="ST_VerticalJc" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocGrid">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="lines"/>
            <xsd:enumeration value="linesAndChars"/>
            <xsd:enumeration value="snapToChars"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocGrid">
        <xsd:attribute name="type" type="ST_DocGrid"/>
        <xsd:attribute name="linePitch" type="ST_DecimalNumber"/>
        <xsd:attribute name="charSpace" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HdrFtr">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="even"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="first"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FtnEdn">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="separator"/>
            <xsd:enumeration value="continuationSeparator"/>
            <xsd:enumeration value="continuationNotice"/>
        </xsd:restriction>
    </xsd:simpleType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_HdrFtrRef">
        <xsd:complexContent>
            <xsd:extension base="CT_Rel">
                <xsd:attribute name="type" type="ST_HdrFtr" use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:group name="EG_HdrFtrReferences">
        <xsd:choice>
            <xsd:element name="headerReference" type="CT_HdrFtrRef" minOccurs="0"/>
            <xsd:element name="footerReference" type="CT_HdrFtrRef" minOccurs="0"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_HdrFtr">
        <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_SectPrContents">
        <xsd:sequence>
            <xsd:element name="footnotePr" type="CT_FtnProps" minOccurs="0"/>
            <xsd:element name="endnotePr" type="CT_EdnProps" minOccurs="0"/>
            <xsd:element name="type" type="CT_SectType" minOccurs="0"/>
            <xsd:element name="pgSz" type="CT_PageSz" minOccurs="0"/>
            <xsd:element name="pgMar" type="CT_PageMar" minOccurs="0"/>
            <xsd:element name="paperSrc" type="CT_PaperSource" minOccurs="0"/>
            <xsd:element name="pgBorders" type="CT_PageBorders" minOccurs="0"/>
            <xsd:element name="lnNumType" type="CT_LineNumber" minOccurs="0"/>
            <xsd:element name="pgNumType" type="CT_PageNumber" minOccurs="0"/>
            <xsd:element name="cols" type="CT_Columns" minOccurs="0"/>
            <xsd:element name="formProt" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="vAlign" type="CT_VerticalJc" minOccurs="0"/>
            <xsd:element name="noEndnote" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="titlePg" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0"/>
            <xsd:element name="bidi" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rtlGutter" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="docGrid" type="CT_DocGrid" minOccurs="0"/>
            <xsd:element name="printerSettings" type="CT_Rel" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="AG_SectPrAttributes">
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidSect" type="ST_LongHexNumber"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_SectPrBase">
        <xsd:sequence>
            <xsd:group ref="EG_SectPrContents" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_SectPrAttributes"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SectPr">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:group ref="EG_HdrFtrReferences" minOccurs="0" maxOccurs="6"/>
            <xsd:group ref="EG_SectPrContents" minOccurs="0"/>
            <xsd:element name="sectPrChange" type="CT_SectPrChange" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_SectPrAttributes"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BrType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="page"/>
            <xsd:enumeration value="column"/>
            <xsd:enumeration value="textWrapping"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BrClear">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Br">
        <xsd:attribute name="type" type="ST_BrType" use="optional"/>
        <xsd:attribute name="clear" type="ST_BrClear" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PTabAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PTabRelativeTo">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="margin"/>
            <xsd:enumeration value="indent"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PTabLeader">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="dot"/>
            <xsd:enumeration value="hyphen"/>
            <xsd:enumeration value="underscore"/>
            <xsd:enumeration value="middleDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PTab">
        <xsd:attribute name="alignment" type="ST_PTabAlignment" use="required"/>
        <xsd:attribute name="relativeTo" type="ST_PTabRelativeTo" use="required"/>
        <xsd:attribute name="leader" type="ST_PTabLeader" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Sym">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="font" type="s:ST_String"/>
        <xsd:attribute name="char" type="ST_ShortHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ProofErr">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="spellStart"/>
            <xsd:enumeration value="spellEnd"/>
            <xsd:enumeration value="gramStart"/>
            <xsd:enumeration value="gramEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ProofErr">
        <xsd:attribute name="type" type="ST_ProofErr" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EdGrp">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="everyone"/>
            <xsd:enumeration value="administrators"/>
            <xsd:enumeration value="contributors"/>
            <xsd:enumeration value="editors"/>
            <xsd:enumeration value="owners"/>
            <xsd:enumeration value="current"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Perm">
        <xsd:attribute name="id" type="s:ST_String" use="required"/>
        <xsd:attribute name="displacedByCustomXml" type="ST_DisplacedByCustomXml" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PermStart">
        <xsd:complexContent>
            <xsd:extension base="CT_Perm">
                <xsd:attribute name="edGrp" type="ST_EdGrp" use="optional"/>
                <xsd:attribute name="ed" type="s:ST_String" use="optional"/>
                <xsd:attribute name="colFirst" type="ST_DecimalNumber" use="optional"/>
                <xsd:attribute name="colLast" type="ST_DecimalNumber" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Text">
        <xsd:simpleContent>
            <xsd:extension base="s:ST_String">
                <xsd:attribute ref="xml:space" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:group name="EG_RunInnerContent">
        <xsd:choice>
            <xsd:element name="br" type="CT_Br"/>
            <xsd:element name="t" type="CT_Text"/>
            <xsd:element name="contentPart" type="CT_Rel"/>
            <xsd:element name="delText" type="CT_Text"/>
            <xsd:element name="instrText" type="CT_Text"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="delInstrText" type="CT_Text"/>
            <xsd:element name="noBreakHyphen" type="CT_Empty"/>
            <xsd:element name="softHyphen" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="dayShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="monthShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="yearShort" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="dayLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="monthLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="yearLong" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="annotationRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="footnoteRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="endnoteRef" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="separator" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="continuationSeparator" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="sym" type="CT_Sym" minOccurs="0"/>
            <xsd:element name="pgNum" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="cr" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="tab" type="CT_Empty" minOccurs="0"/>
            <xsd:element name="object" type="CT_Object"/>
            <xsd:element name="fldChar" type="CT_FldChar"/>
            <xsd:element name="ruby" type="CT_Ruby"/>
            <xsd:element name="footnoteReference" type="CT_FtnEdnRef"/>
            <xsd:element name="endnoteReference" type="CT_FtnEdnRef"/>
            <xsd:element name="commentReference" type="CT_Markup"/>
            <xsd:element name="drawing" type="CT_Drawing"/>
            <xsd:element name="ptab" type="CT_PTab" minOccurs="0"/>
            <xsd:element name="lastRenderedPageBreak" type="CT_Empty" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_R">
        <xsd:sequence>
            <xsd:group ref="EG_RPr" minOccurs="0"/>
            <xsd:group ref="EG_RunInnerContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Hint">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="eastAsia"/>
            <xsd:enumeration value="cs"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Theme">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="majorEastAsia"/>
            <xsd:enumeration value="majorBidi"/>
            <xsd:enumeration value="majorAscii"/>
            <xsd:enumeration value="majorHAnsi"/>
            <xsd:enumeration value="minorEastAsia"/>
            <xsd:enumeration value="minorBidi"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="minorAscii"/>
            <xsd:enumeration value="minorHAnsi"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Fonts">
        <xsd:attribute name="hint" type="ST_Hint"/>
        <xsd:attribute name="ascii" type="s:ST_String"/>
        <xsd:attribute name="hAnsi" type="s:ST_String"/>
        <xsd:attribute name="eastAsia" type="s:ST_String"/>
        <xsd:attribute name="cs" type="s:ST_String"/>
        <xsd:attribute name="asciiTheme" type="ST_Theme"/>
        <xsd:attribute name="hAnsiTheme" type="ST_Theme"/>
        <xsd:attribute name="eastAsiaTheme" type="ST_Theme"/>
        <xsd:attribute name="cstheme" type="ST_Theme"/>
    </xsd:complexType>
    <xsd:group name="EG_RPrBase">
        <xsd:choice>
            <xsd:element name="rStyle" type="CT_String"/>
            <xsd:element name="rFonts" type="CT_Fonts"/>
            <xsd:element name="b" type="CT_OnOff"/>
            <xsd:element name="bCs" type="CT_OnOff"/>
            <xsd:element name="i" type="CT_OnOff"/>
            <xsd:element name="iCs" type="CT_OnOff"/>
            <xsd:element name="caps" type="CT_OnOff"/>
            <xsd:element name="smallCaps" type="CT_OnOff"/>
            <xsd:element name="strike" type="CT_OnOff"/>
            <xsd:element name="dstrike" type="CT_OnOff"/>
            <xsd:element name="outline" type="CT_OnOff"/>
            <xsd:element name="shadow" type="CT_OnOff"/>
            <xsd:element name="emboss" type="CT_OnOff"/>
            <xsd:element name="imprint" type="CT_OnOff"/>
            <xsd:element name="noProof" type="CT_OnOff"/>
            <xsd:element name="snapToGrid" type="CT_OnOff"/>
            <xsd:element name="vanish" type="CT_OnOff"/>
            <xsd:element name="webHidden" type="CT_OnOff"/>
            <xsd:element name="color" type="CT_Color"/>
            <xsd:element name="spacing" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="w" type="CT_TextScale"/>
            <xsd:element name="kern" type="CT_HpsMeasure"/>
            <xsd:element name="position" type="CT_SignedHpsMeasure"/>
            <xsd:element name="sz" type="CT_HpsMeasure"/>
            <xsd:element name="szCs" type="CT_HpsMeasure"/>
            <xsd:element name="highlight" type="CT_Highlight"/>
            <xsd:element name="u" type="CT_Underline"/>
            <xsd:element name="effect" type="CT_TextEffect"/>
            <xsd:element name="bdr" type="CT_Border"/>
            <xsd:element name="shd" type="CT_Shd"/>
            <xsd:element name="fitText" type="CT_FitText"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignRun"/>
            <xsd:element name="rtl" type="CT_OnOff"/>
            <xsd:element name="cs" type="CT_OnOff"/>
            <xsd:element name="em" type="CT_Em"/>
            <xsd:element name="lang" type="CT_Language"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="eastAsianLayout" type="CT_EastAsianLayout"/>
            <xsd:element name="specVanish" type="CT_OnOff"/>
            <xsd:element name="oMath" type="CT_OnOff"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_RPrContent">
        <xsd:sequence>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rPrChange" type="CT_RPrChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_RPr">
        <xsd:sequence>
            <xsd:group ref="EG_RPrContent" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_RPr">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="EG_RPrMath">
        <xsd:choice>
            <xsd:group ref="EG_RPr"/>
            <xsd:element name="ins" type="CT_MathCtrlIns"/>
            <xsd:element name="del" type="CT_MathCtrlDel"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_MathCtrlIns">
      <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
          <xsd:choice minOccurs="0">
            <xsd:element name="del" type="CT_RPrChange" minOccurs="1"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="1"/>
          </xsd:choice>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_MathCtrlDel">
      <xsd:complexContent>
        <xsd:extension base="CT_TrackChange">
          <xsd:choice minOccurs="0">
            <xsd:element name="rPr" type="CT_RPr" minOccurs="1"/>
          </xsd:choice>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrOriginal">
        <xsd:sequence>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ParaRPrOriginal">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:sequence>
            <xsd:group ref="EG_ParaRPrTrackChanges" minOccurs="0"/>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ParaRPr">
        <xsd:sequence>
            <xsd:group ref="EG_ParaRPrTrackChanges" minOccurs="0"/>
            <xsd:group ref="EG_RPrBase" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rPrChange" type="CT_ParaRPrChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_ParaRPrTrackChanges">
        <xsd:sequence>
            <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="del" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="moveFrom" type="CT_TrackChange" minOccurs="0"/>
            <xsd:element name="moveTo" type="CT_TrackChange" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_AltChunk">
        <xsd:sequence>
            <xsd:element name="altChunkPr" type="CT_AltChunkPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AltChunkPr">
        <xsd:sequence>
            <xsd:element name="matchSrc" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_RubyAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="distributeLetter"/>
            <xsd:enumeration value="distributeSpace"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="rightVertical"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_RubyAlign">
        <xsd:attribute name="val" type="ST_RubyAlign" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RubyPr">
        <xsd:sequence>
            <xsd:element name="rubyAlign" type="CT_RubyAlign"/>
            <xsd:element name="hps" type="CT_HpsMeasure"/>
            <xsd:element name="hpsRaise" type="CT_HpsMeasure"/>
            <xsd:element name="hpsBaseText" type="CT_HpsMeasure"/>
            <xsd:element name="lid" type="CT_Lang"/>
            <xsd:element name="dirty" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:group name="EG_RubyContent">
        <xsd:choice>
            <xsd:element name="r" type="CT_R"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_RubyContent">
        <xsd:group ref="EG_RubyContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ruby">
        <xsd:sequence>
            <xsd:element name="rubyPr" type="CT_RubyPr"/>
            <xsd:element name="rt" type="CT_RubyContent"/>
            <xsd:element name="rubyBase" type="CT_RubyContent"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Lock">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sdtLocked"/>
            <xsd:enumeration value="contentLocked"/>
            <xsd:enumeration value="unlocked"/>
            <xsd:enumeration value="sdtContentLocked"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Lock">
        <xsd:attribute name="val" type="ST_Lock"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtListItem">
        <xsd:attribute name="displayText" type="s:ST_String"/>
        <xsd:attribute name="value" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SdtDateMappingType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="dateTime"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SdtDateMappingType">
        <xsd:attribute name="val" type="ST_SdtDateMappingType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalendarType">
        <xsd:attribute name="val" type="s:ST_CalendarType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDate">
        <xsd:sequence>
            <xsd:element name="dateFormat" type="CT_String" minOccurs="0"/>
            <xsd:element name="lid" type="CT_Lang" minOccurs="0"/>
            <xsd:element name="storeMappedDataAs" type="CT_SdtDateMappingType" minOccurs="0"/>
            <xsd:element name="calendar" type="CT_CalendarType" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="fullDate" type="ST_DateTime" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:complexType name="CT_SdtComboBox">
        <xsd:sequence>
            <xsd:element name="listItem" type="CT_SdtListItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="lastValue" type="s:ST_String" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDocPart">
        <xsd:sequence>
            <xsd:element name="docPartGallery" type="CT_String" minOccurs="0"/>
            <xsd:element name="docPartCategory" type="CT_String" minOccurs="0"/>
            <xsd:element name="docPartUnique" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtDropDownList">
        <xsd:sequence>
            <xsd:element name="listItem" type="CT_SdtListItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="lastValue" type="s:ST_String" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_Placeholder">
        <xsd:sequence>
            <xsd:element name="docPart" type="CT_String"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtText">
        <xsd:attribute name="multiLine" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBinding">
        <xsd:attribute name="prefixMappings" type="s:ST_String"/>
        <xsd:attribute name="xpath" type="s:ST_String" use="required"/>
        <xsd:attribute name="storeItemID" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtPr">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
            <xsd:element name="alias" type="CT_String" minOccurs="0"/>
            <xsd:element name="tag" type="CT_String" minOccurs="0"/>
            <xsd:element name="id" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lock" type="CT_Lock" minOccurs="0"/>
            <xsd:element name="placeholder" type="CT_Placeholder" minOccurs="0"/>
            <xsd:element name="temporary" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="showingPlcHdr" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="dataBinding" type="CT_DataBinding" minOccurs="0"/>
            <xsd:element name="label" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="tabIndex" type="CT_UnsignedDecimalNumber" minOccurs="0"/>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="equation" type="CT_Empty"/>
                <xsd:element name="comboBox" type="CT_SdtComboBox"/>
                <xsd:element name="date" type="CT_SdtDate"/>
                <xsd:element name="docPartObj" type="CT_SdtDocPart"/>
                <xsd:element name="docPartList" type="CT_SdtDocPart"/>
                <xsd:element name="dropDownList" type="CT_SdtDropDownList"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="picture" type="CT_Empty"/>
            <xsd:element name="richText" type="CT_Empty"/>
            <xsd:element name="text" type="CT_SdtText"/>
            <xsd:element name="citation" type="CT_Empty"/>
            <xsd:element name="group" type="CT_Empty"/>
            <xsd:element name="bibliography" type="CT_Empty"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CT_SdtEndPr">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
    </xsd:choice>
</xsd:complexType>
<xsd:group name="EG_ContentRunContent">
    <xsd:choice>
        <xsd:element name="customXml" type="CT_CustomXmlRun"/>
        <xsd:element name="smartTag" type="CT_SmartTagRun"/>
        <xsd:element name="sdt" type="CT_SdtRun"/>
        <xsd:element name="dir" type="CT_DirContentRun"/>
        <xsd:element name="bdo" type="CT_BdoContentRun"/>
        <xsd:element name="r" type="CT_R"/>
        <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="CT_DirContentRun">
    <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:attribute name="val" type="ST_Direction" use="optional"/>
</xsd:complexType>
<xsd:complexType name="CT_BdoContentRun">
    <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:attribute name="val" type="ST_Direction" use="optional"/>
</xsd:complexType>
<xsd:simpleType name="ST_Direction">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ltr"/>
        <xsd:enumeration value="rtl"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CT_SdtContentRun">
    <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
</xsd:complexType>
<xsd:group name="EG_ContentBlockContent">
    <xsd:choice>
        <xsd:element name="customXml" type="CT_CustomXmlBlock"/>
        <xsd:element name="sdt" type="CT_SdtBlock"/>
        <xsd:element name="p" type="CT_P" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="tbl" type="CT_Tbl" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
</xsd:group>
<xsd:complexType name="CT_SdtContentBlock">
    <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:group name="EG_ContentRowContent">
        <xsd:choice>
            <xsd:element name="tr" type="CT_Row" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="customXml" type="CT_CustomXmlRow"/>
            <xsd:element name="sdt" type="CT_SdtRow"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_SdtContentRow">
        <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:group name="EG_ContentCellContent">
        <xsd:choice>
            <xsd:element name="tc" type="CT_Tc" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="customXml" type="CT_CustomXmlCell"/>
            <xsd:element name="sdt" type="CT_SdtCell"/>
            <xsd:group ref="EG_RunLevelElts" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_SdtContentCell">
        <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtBlock">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentBlock" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtRun">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentRun" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtCell">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentCell" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SdtRow">
        <xsd:sequence>
            <xsd:element name="sdtPr" type="CT_SdtPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtEndPr" type="CT_SdtEndPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sdtContent" type="CT_SdtContentRow" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Attr">
        <xsd:attribute name="uri" type="s:ST_String"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlRun">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagRun">
        <xsd:sequence>
            <xsd:element name="smartTagPr" type="CT_SmartTagPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlBlock">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlPr">
        <xsd:sequence>
            <xsd:element name="placeholder" type="CT_String" minOccurs="0"/>
            <xsd:element name="attr" type="CT_Attr" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlRow">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomXmlCell">
        <xsd:sequence>
            <xsd:element name="customXmlPr" type="CT_CustomXmlPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="element" type="s:ST_XmlName" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagPr">
        <xsd:sequence>
            <xsd:element name="attr" type="CT_Attr" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:group name="EG_PContent">
        <xsd:choice>
            <xsd:group ref="EG_ContentRunContent" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="fldSimple" type="CT_SimpleField" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="hyperlink" type="CT_Hyperlink"/>
            <xsd:element name="subDoc" type="CT_Rel"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_P">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPr" minOccurs="0"/>
            <xsd:group ref="EG_PContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidP" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidRDefault" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblWidth">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="pct"/>
            <xsd:enumeration value="dxa"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Height">
        <xsd:attribute name="val" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="hRule" type="ST_HeightRule"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MeasurementOrPercent">
        <xsd:union memberTypes="ST_DecimalNumberOrPercent s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblWidth">
        <xsd:attribute name="w" type="ST_MeasurementOrPercent"/>
        <xsd:attribute name="type" type="ST_TblWidth"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridCol">
        <xsd:attribute name="w" type="s:ST_TwipsMeasure"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGridBase">
        <xsd:sequence>
            <xsd:element name="gridCol" type="CT_TblGridCol" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblGrid">
        <xsd:complexContent>
            <xsd:extension base="CT_TblGridBase">
                <xsd:sequence>
                    <xsd:element name="tblGridChange" type="CT_TblGridChange" minOccurs="0"/>
                </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TcBorders">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="start" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideH" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideV" type="CT_Border" minOccurs="0"/>
            <xsd:element name="tl2br" type="CT_Border" minOccurs="0"/>
            <xsd:element name="tr2bl" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TcMar">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="start" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_Merge">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="continue"/>
            <xsd:enumeration value="restart"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_VMerge">
        <xsd:attribute name="val" type="ST_Merge"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPrBase">
        <xsd:sequence>
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gridSpan" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="vMerge" type="CT_VMerge" minOccurs="0"/>
            <xsd:element name="tcBorders" type="CT_TcBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0"/>
            <xsd:element name="noWrap" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="tcMar" type="CT_TcMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="textDirection" type="CT_TextDirection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcFitText" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vAlign" type="CT_VerticalJc" minOccurs="0"/>
            <xsd:element name="hideMark" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="headers" type="CT_Headers" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TcPrInner">
                <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
                    <xsd:element name="tcPrChange" type="CT_TcPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TcPrInner">
        <xsd:complexContent>
            <xsd:extension base="CT_TcPrBase">
                <xsd:sequence>
                    <xsd:group ref="EG_CellMarkupElements" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Tc">
        <xsd:sequence>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cnf">
        <xsd:attribute name="firstRow" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRow" type="s:ST_OnOff"/>
        <xsd:attribute name="firstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="oddVBand" type="s:ST_OnOff"/>
        <xsd:attribute name="evenVBand" type="s:ST_OnOff"/>
        <xsd:attribute name="oddHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="evenHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="firstRowFirstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="firstRowLastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRowFirstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRowLastColumn" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Headers">
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="header" type="CT_String"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPrBase">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="cnfStyle" type="CT_Cnf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="divId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="gridBefore" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="gridAfter" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="wBefore" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="wAfter" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cantSplit" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="trHeight" type="CT_Height" minOccurs="0"/>
            <xsd:element name="tblHeader" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="hidden" type="CT_OnOff" minOccurs="0"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_TrPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TrPrBase">
                <xsd:sequence>
                    <xsd:element name="ins" type="CT_TrackChange" minOccurs="0"/>
                    <xsd:element name="del" type="CT_TrackChange" minOccurs="0"/>
                    <xsd:element name="trPrChange" type="CT_TrPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Row">
        <xsd:sequence>
            <xsd:element name="tblPrEx" type="CT_TblPrEx" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ContentCellContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="rsidRPr" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidR" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidDel" type="ST_LongHexNumber"/>
        <xsd:attribute name="rsidTr" type="ST_LongHexNumber"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblLayoutType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="fixed"/>
            <xsd:enumeration value="autofit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblLayoutType">
        <xsd:attribute name="type" type="ST_TblLayoutType"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblOverlap">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="never"/>
            <xsd:enumeration value="overlap"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblOverlap">
        <xsd:attribute name="val" type="ST_TblOverlap" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPPr">
        <xsd:attribute name="leftFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="rightFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="topFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="bottomFromText" type="s:ST_TwipsMeasure"/>
        <xsd:attribute name="vertAnchor" type="ST_VAnchor"/>
        <xsd:attribute name="horzAnchor" type="ST_HAnchor"/>
        <xsd:attribute name="tblpXSpec" type="s:ST_XAlign"/>
        <xsd:attribute name="tblpX" type="ST_SignedTwipsMeasure"/>
        <xsd:attribute name="tblpYSpec" type="s:ST_YAlign"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="tblpY" type="ST_SignedTwipsMeasure"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TblCellMar">
        <xsd:sequence>
            <xsd:element name="top" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="start" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bottom" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblBorders">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="start" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="end" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideH" type="CT_Border" minOccurs="0"/>
            <xsd:element name="insideV" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrBase">
        <xsd:sequence>
            <xsd:element name="tblStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="tblpPr" type="CT_TblPPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblOverlap" type="CT_TblOverlap" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bidiVisual" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblStyleRowBandSize" type="CT_DecimalNumber" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tblStyleColBandSize" type="CT_DecimalNumber" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tblW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblInd" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblBorders" type="CT_TblBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLayout" type="CT_TblLayoutType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellMar" type="CT_TblCellMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLook" type="CT_TblLook" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCaption" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblDescription" type="CT_String" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPr">
        <xsd:complexContent>
            <xsd:extension base="CT_TblPrBase">
                <xsd:sequence>
                    <xsd:element name="tblPrChange" type="CT_TblPrChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrExBase">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="tblW" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="jc" type="CT_JcTable" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellSpacing" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblInd" type="CT_TblWidth" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblBorders" type="CT_TblBorders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shd" type="CT_Shd" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLayout" type="CT_TblLayoutType" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblCellMar" type="CT_TblCellMar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblLook" type="CT_TblLook" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblPrEx">
        <xsd:complexContent>
            <xsd:extension base="CT_TblPrExBase">
                <xsd:sequence>
                    <xsd:element name="tblPrExChange" type="CT_TblPrExChange" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Tbl">
        <xsd:sequence>
            <xsd:group ref="EG_RangeMarkupElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tblPr" type="CT_TblPr"/>
            <xsd:element name="tblGrid" type="CT_TblGrid"/>
            <xsd:group ref="EG_ContentRowContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TblLook">
        <xsd:attribute name="firstRow" type="s:ST_OnOff"/>
        <xsd:attribute name="lastRow" type="s:ST_OnOff"/>
        <xsd:attribute name="firstColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="lastColumn" type="s:ST_OnOff"/>
        <xsd:attribute name="noHBand" type="s:ST_OnOff"/>
        <xsd:attribute name="noVBand" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FtnPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="pageBottom"/>
            <xsd:enumeration value="beneathText"/>
            <xsd:enumeration value="sectEnd"/>
            <xsd:enumeration value="docEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FtnPos">
        <xsd:attribute name="val" type="ST_FtnPos" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_EdnPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sectEnd"/>
            <xsd:enumeration value="docEnd"/>
        </xsd:restriction>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:simpleType>
    <xsd:complexType name="CT_EdnPos">
        <xsd:attribute name="val" type="ST_EdnPos" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumFmt">
        <xsd:attribute name="val" type="ST_NumberFormat" use="required"/>
        <xsd:attribute name="format" type="s:ST_String" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RestartNumber">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="continuous"/>
            <xsd:enumeration value="eachSect"/>
            <xsd:enumeration value="eachPage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NumRestart">
        <xsd:attribute name="val" type="ST_RestartNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdnRef">
        <xsd:attribute name="customMarkFollows" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="id" use="required" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdnSepRef">
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnEdn">
        <xsd:sequence>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_FtnEdn" use="optional"/>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_FtnEdnNumProps">
        <xsd:sequence>
            <xsd:element name="numStart" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numRestart" type="CT_NumRestart" minOccurs="0"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_FtnProps">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_FtnPos" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:group ref="EG_FtnEdnNumProps" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EdnProps">
        <xsd:sequence>
            <xsd:element name="pos" type="CT_EdnPos" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:group ref="EG_FtnEdnNumProps" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FtnDocProps">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:complexContent>
            <xsd:extension base="CT_FtnProps">
                <xsd:sequence>
                    <xsd:element name="footnote" type="CT_FtnEdnSepRef" minOccurs="0" maxOccurs="3"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_EdnDocProps">
        <xsd:complexContent>
            <xsd:extension base="CT_EdnProps">
                <xsd:sequence>
                    <xsd:element name="endnote" type="CT_FtnEdnSepRef" minOccurs="0" maxOccurs="3"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_RecipientData">
        <xsd:sequence>
            <xsd:element name="active" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="column" type="CT_DecimalNumber" minOccurs="1"/>
            <xsd:element name="uniqueTag" type="CT_Base64Binary" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Base64Binary">
        <xsd:attribute name="val" type="xsd:base64Binary" use="required">
        </xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="CT_Recipients">
        <xsd:sequence>
            <xsd:element name="recipientData" type="CT_RecipientData" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="recipients" type="CT_Recipients"/>
    <xsd:complexType name="CT_OdsoFieldMapData">
        <xsd:sequence>
            <xsd:element name="type" type="CT_MailMergeOdsoFMDFieldType" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="mappedName" type="CT_String" minOccurs="0"/>
            <xsd:element name="column" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lid" type="CT_Lang" minOccurs="0"/>
            <xsd:element name="dynamicAddress" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_MailMergeSourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="database"/>
            <xsd:enumeration value="addressBook"/>
            <xsd:enumeration value="document1"/>
            <xsd:enumeration value="document2"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="email"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="native"/>
            <xsd:enumeration value="legacy"/>
            <xsd:enumeration value="master"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MailMergeSourceType">
        <xsd:attribute name="val" use="required" type="ST_MailMergeSourceType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Odso">
        <xsd:sequence>
            <xsd:element name="udl" type="CT_String" minOccurs="0"/>
            <xsd:element name="table" type="CT_String" minOccurs="0"/>
            <xsd:element name="src" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="colDelim" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="type" type="CT_MailMergeSourceType" minOccurs="0"/>
            <xsd:element name="fHdr" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="fieldMapData" type="CT_OdsoFieldMapData" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="recipientData" type="CT_Rel" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MailMerge">
        <xsd:sequence>
            <xsd:element name="mainDocumentType" type="CT_MailMergeDocType" minOccurs="1"/>
            <xsd:element name="linkToQuery" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="dataType" type="CT_MailMergeDataType" minOccurs="1"/>
            <xsd:element name="connectString" type="CT_String" minOccurs="0"/>
            <xsd:element name="query" type="CT_String" minOccurs="0"/>
            <xsd:element name="dataSource" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="headerSource" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="doNotSuppressBlankLines" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="destination" type="CT_MailMergeDest" minOccurs="0"/>
            <xsd:element name="addressFieldName" type="CT_String" minOccurs="0"/>
            <xsd:element name="mailSubject" type="CT_String" minOccurs="0"/>
            <xsd:element name="mailAsAttachment" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="viewMergedData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="activeRecord" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="checkErrors" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="odso" type="CT_Odso" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TargetScreenSz">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="544x376"/>
            <xsd:enumeration value="640x480"/>
            <xsd:enumeration value="720x512"/>
            <xsd:enumeration value="800x600"/>
            <xsd:enumeration value="1024x768"/>
            <xsd:enumeration value="1152x882"/>
            <xsd:enumeration value="1152x900"/>
            <xsd:enumeration value="1280x1024"/>
            <xsd:enumeration value="1600x1200"/>
            <xsd:enumeration value="1800x1440"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="1920x1200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TargetScreenSz">
        <xsd:attribute name="val" type="ST_TargetScreenSz" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Compat">
        <xsd:sequence>
            <xsd:element name="spaceForUL" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="balanceSingleByteDoubleByteWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotLeaveBackslashAlone" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ulTrailSpace" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotExpandShiftReturn" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="adjustLineHeightInTable" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="applyBreakingRules" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="compatSetting" type="CT_CompatSetting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CompatSetting">
        <xsd:attribute name="name" type="s:ST_String"/>
        <xsd:attribute name="uri" type="s:ST_String"/>
        <xsd:attribute name="val" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocVar">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocVars">
        <xsd:sequence>
            <xsd:element name="docVar" type="CT_DocVar" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocRsids">
        <xsd:sequence>
            <xsd:element name="rsidRoot" type="CT_LongHexNumber" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rsid" type="CT_LongHexNumber" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_CharacterSpacing">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="doNotCompress"/>
            <xsd:enumeration value="compressPunctuation"/>
            <xsd:enumeration value="compressPunctuationAndJapaneseKana"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_CharacterSpacing">
        <xsd:attribute name="val" type="ST_CharacterSpacing" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SaveThroughXslt">
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="solutionID" type="s:ST_String" use="optional"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_RPrDefault">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PPrDefault">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocDefaults">
        <xsd:sequence>
            <xsd:element name="rPrDefault" type="CT_RPrDefault" minOccurs="0"/>
            <xsd:element name="pPrDefault" type="CT_PPrDefault" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_WmlColorSchemeIndex">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="dark1"/>
            <xsd:enumeration value="light1"/>
            <xsd:enumeration value="dark2"/>
            <xsd:enumeration value="light2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hyperlink"/>
            <xsd:enumeration value="followedHyperlink"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ColorSchemeMapping">
        <xsd:attribute name="bg1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="t1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="bg2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="t2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent1" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent2" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent3" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent4" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent5" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="accent6" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="hyperlink" type="ST_WmlColorSchemeIndex"/>
        <xsd:attribute name="followedHyperlink" type="ST_WmlColorSchemeIndex"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ReadingModeInkLockDown">
        <xsd:attribute name="actualPg" type="s:ST_OnOff" use="required"/>
        <xsd:attribute name="w" type="ST_PixelsMeasure" use="required"/>
        <xsd:attribute name="h" type="ST_PixelsMeasure" use="required"/>
        <xsd:attribute name="fontSz" type="ST_DecimalNumberOrPercent" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WriteProtection">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="recommended" type="s:ST_OnOff" use="optional"/>
        <xsd:attributeGroup ref="AG_Password"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Settings">
        <xsd:sequence>
            <xsd:element name="writeProtection" type="CT_WriteProtection" minOccurs="0"/>
            <xsd:element name="view" type="CT_View" minOccurs="0"/>
            <xsd:element name="zoom" type="CT_Zoom" minOccurs="0"/>
            <xsd:element name="removePersonalInformation" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="removeDateAndTime" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotDisplayPageBoundaries" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="displayBackgroundShape" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printPostScriptOverText" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printFractionalCharacterWidth" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="printFormsData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="embedTrueTypeFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="embedSystemFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveSubsetFonts" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveFormsData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="mirrorMargins" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alignBordersAndEdges" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bordersDoNotSurroundHeader" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bordersDoNotSurroundFooter" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="gutterAtTop" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideSpellingErrors" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hideGrammaticalErrors" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="activeWritingStyle" type="CT_WritingStyle" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="proofState" type="CT_Proof" minOccurs="0"/>
            <xsd:element name="formsDesign" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="attachedTemplate" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="linkStyles" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="stylePaneFormatFilter" type="CT_StylePaneFilter" minOccurs="0"/>
            <xsd:element name="stylePaneSortMethod" type="CT_StyleSort" minOccurs="0"/>
            <xsd:element name="documentType" type="CT_DocType" minOccurs="0"/>
            <xsd:element name="mailMerge" type="CT_MailMerge" minOccurs="0"/>
            <xsd:element name="revisionView" type="CT_TrackChangesView" minOccurs="0"/>
            <xsd:element name="trackRevisions" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotTrackMoves" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotTrackFormatting" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="documentProtection" type="CT_DocProtect" minOccurs="0"/>
            <xsd:element name="autoFormatOverride" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="styleLockTheme" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="styleLockQFSet" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="defaultTabStop" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="autoHyphenation" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="consecutiveHyphenLimit" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="hyphenationZone" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="doNotHyphenateCaps" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="showEnvelope" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="summaryLength" type="CT_DecimalNumberOrPrecent" minOccurs="0"/>
            <xsd:element name="clickAndTypeStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="defaultTableStyle" type="CT_String" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="evenAndOddHeaders" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bookFoldRevPrinting" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bookFoldPrinting" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bookFoldPrintingSheets" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="drawingGridHorizontalSpacing" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="drawingGridVerticalSpacing" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="displayHorizontalDrawingGridEvery" type="CT_DecimalNumber"
              minOccurs="0"/>
            <xsd:element name="displayVerticalDrawingGridEvery" type="CT_DecimalNumber"
              minOccurs="0"/>
            <xsd:element name="doNotUseMarginsForDrawingGridOrigin" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="drawingGridHorizontalOrigin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="drawingGridVerticalOrigin" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="doNotShadeFormData" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noPunctuationKerning" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="characterSpacingControl" type="CT_CharacterSpacing" minOccurs="0"/>
            <xsd:element name="printTwoOnOne" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="strictFirstAndLastChars" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="noLineBreaksAfter" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="noLineBreaksBefore" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="savePreviewPicture" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotValidateAgainstSchema" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveInvalidXml" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="ignoreMixedContent" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alwaysShowPlaceholderText" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotDemarcateInvalidXml" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveXmlDataOnly" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="useXSLTWhenSaving" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="saveThroughXslt" type="CT_SaveThroughXslt" minOccurs="0"/>
            <xsd:element name="showXMLTags" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="alwaysMergeEmptyNamespace" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="updateFields" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="footnotePr" type="CT_FtnDocProps" minOccurs="0"/>
            <xsd:element name="endnotePr" type="CT_EdnDocProps" minOccurs="0"/>
            <xsd:element name="compat" type="CT_Compat" minOccurs="0"/>
            <xsd:element name="docVars" type="CT_DocVars" minOccurs="0"/>
            <xsd:element name="rsids" type="CT_DocRsids" minOccurs="0"/>
            <xsd:element ref="m:mathPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="attachedSchema" type="CT_String" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="themeFontLang" type="CT_Language" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="clrSchemeMapping" type="CT_ColorSchemeMapping" minOccurs="0"/>
            <xsd:element name="doNotIncludeSubdocsInStats" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotAutoCompressPictures" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="forceUpgrade" type="CT_Empty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="captions" type="CT_Captions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="readModeInkLockDown" type="CT_ReadingModeInkLockDown" minOccurs="0"/>
            <xsd:element name="smartTagType" type="CT_SmartTagType" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element ref="sl:schemaLibrary" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="doNotEmbedSmartTags" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="decimalSymbol" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="listSeparator" type="CT_String" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_StyleSort">
        <xsd:attribute name="val" type="ST_StyleSort" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StylePaneFilter">
        <xsd:attribute name="allStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="customStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="latentStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="stylesInUse" type="s:ST_OnOff"/>
        <xsd:attribute name="headingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="numberingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="tableStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnRuns" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnParagraphs" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnNumbering" type="s:ST_OnOff"/>
        <xsd:attribute name="directFormattingOnTables" type="s:ST_OnOff"/>
        <xsd:attribute name="clearFormatting" type="s:ST_OnOff"/>
        <xsd:attribute name="top3HeadingStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="visibleStyles" type="s:ST_OnOff"/>
        <xsd:attribute name="alternateStyleNames" type="s:ST_OnOff"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_StyleSort">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="name"/>
            <xsd:enumeration value="priority"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="font"/>
            <xsd:enumeration value="basedOn"/>
            <xsd:enumeration value="type"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WebSettings">
        <xsd:sequence>
            <xsd:element name="frameset" type="CT_Frameset" minOccurs="0"/>
            <xsd:element name="divs" type="CT_Divs" minOccurs="0"/>
            <xsd:element name="encoding" type="CT_String" minOccurs="0"/>
            <xsd:element name="optimizeForBrowser" type="CT_OptimizeForBrowser" minOccurs="0"/>
            <xsd:element name="allowPNG" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotRelyOnCSS" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotSaveAsSingleFile" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotOrganizeInFolder" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="doNotUseLongFileNames" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="pixelsPerInch" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="targetScreenSz" type="CT_TargetScreenSz" minOccurs="0"/>
            <xsd:element name="saveSmartTagsAsXml" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_FrameScrollbar">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:simpleType>
    <xsd:complexType name="CT_FrameScrollbar">
        <xsd:attribute name="val" type="ST_FrameScrollbar" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OptimizeForBrowser">
        <xsd:complexContent>
            <xsd:extension base="CT_OnOff">
                <xsd:attribute name="target" type="s:ST_String" use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Frame">
        <xsd:sequence>
            <xsd:element name="sz" type="CT_String" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="title" type="CT_String" minOccurs="0"/>
            <xsd:element name="longDesc" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="sourceFileName" type="CT_Rel" minOccurs="0"/>
            <xsd:element name="marW" type="CT_PixelsMeasure" minOccurs="0"/>
            <xsd:element name="marH" type="CT_PixelsMeasure" minOccurs="0"/>
            <xsd:element name="scrollbar" type="CT_FrameScrollbar" minOccurs="0"/>
            <xsd:element name="noResizeAllowed" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="linkedToFile" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_FrameLayout">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="rows"/>
            <xsd:enumeration value="cols"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FrameLayout">
        <xsd:attribute name="val" type="ST_FrameLayout" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FramesetSplitbar">
        <xsd:sequence>
            <xsd:element name="w" type="CT_TwipsMeasure" minOccurs="0"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0"/>
            <xsd:element name="noBorder" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="flatBorders" type="CT_OnOff" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Frameset">
        <xsd:sequence>
            <xsd:element name="sz" type="CT_String" minOccurs="0"/>
            <xsd:element name="framesetSplitbar" type="CT_FramesetSplitbar" minOccurs="0"/>
            <xsd:element name="frameLayout" type="CT_FrameLayout" minOccurs="0"/>
            <xsd:element name="title" type="CT_String" minOccurs="0"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="frameset" type="CT_Frameset" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="frame" type="CT_Frame" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:choice>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NumPicBullet">
        <xsd:choice>
            <xsd:element name="drawing" type="CT_Drawing"/>
        </xsd:choice>
        <xsd:attribute name="numPicBulletId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LevelSuffix">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="tab"/>
            <xsd:enumeration value="space"/>
            <xsd:enumeration value="nothing"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LevelSuffix">
        <xsd:attribute name="val" type="ST_LevelSuffix" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LevelText">
        <xsd:attribute name="val" type="s:ST_String" use="optional"/>
        <xsd:attribute name="null" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Lvl">
        <xsd:sequence>
            <xsd:element name="start" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0"/>
            <xsd:element name="lvlRestart" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="pStyle" type="CT_String" minOccurs="0"/>
            <xsd:element name="isLgl" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="suff" type="CT_LevelSuffix" minOccurs="0"/>
            <xsd:element name="lvlText" type="CT_LevelText" minOccurs="0"/>
            <xsd:element name="lvlPicBulletId" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lvlJc" type="CT_Jc" minOccurs="0"/>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="ilvl" type="ST_DecimalNumber" use="required"/>
        <xsd:attribute name="tplc" type="ST_LongHexNumber" use="optional"/>
        <xsd:attribute name="tentative" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MultiLevelType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="singleLevel"/>
            <xsd:enumeration value="multilevel"/>
            <xsd:enumeration value="hybridMultilevel"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MultiLevelType">
        <xsd:attribute name="val" type="ST_MultiLevelType" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AbstractNum">
        <xsd:sequence>
            <xsd:element name="nsid" type="CT_LongHexNumber" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="multiLevelType" type="CT_MultiLevelType" minOccurs="0"/>
            <xsd:element name="tmpl" type="CT_LongHexNumber" minOccurs="0"/>
            <xsd:element name="name" type="CT_String" minOccurs="0"/>
            <xsd:element name="styleLink" type="CT_String" minOccurs="0"/>
            <xsd:element name="numStyleLink" type="CT_String" minOccurs="0"/>
            <xsd:element name="lvl" type="CT_Lvl" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
        <xsd:attribute name="abstractNumId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumLvl">
        <xsd:sequence>
            <xsd:element name="startOverride" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="lvl" type="CT_Lvl" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ilvl" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Num">
        <xsd:sequence>
            <xsd:element name="abstractNumId" type="CT_DecimalNumber" minOccurs="1"/>
            <xsd:element name="lvlOverride" type="CT_NumLvl" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
        <xsd:attribute name="numId" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Numbering">
        <xsd:sequence>
            <xsd:element name="numPicBullet" type="CT_NumPicBullet" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="abstractNum" type="CT_AbstractNum" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="num" type="CT_Num" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="numIdMacAtCleanup" type="CT_DecimalNumber" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TblStyleOverrideType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="wholeTable"/>
            <xsd:enumeration value="firstRow"/>
            <xsd:enumeration value="lastRow"/>
            <xsd:enumeration value="firstCol"/>
            <xsd:enumeration value="lastCol"/>
            <xsd:enumeration value="band1Vert"/>
            <xsd:enumeration value="band2Vert"/>
            <xsd:enumeration value="band1Horz"/>
            <xsd:enumeration value="band2Horz"/>
            <xsd:enumeration value="neCell"/>
            <xsd:enumeration value="nwCell"/>
            <xsd:enumeration value="seCell"/>
            <xsd:enumeration value="swCell"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TblStylePr">
        <xsd:sequence>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0"/>
            <xsd:element name="tblPr" type="CT_TblPrBase" minOccurs="0"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_TblStyleOverrideType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_StyleType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="paragraph"/>
            <xsd:enumeration value="character"/>
            <xsd:enumeration value="table"/>
            <xsd:enumeration value="numbering"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Style">
        <xsd:sequence>
            <xsd:element name="name" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="aliases" type="CT_String" minOccurs="0"/>
            <xsd:element name="basedOn" type="CT_String" minOccurs="0"/>
            <xsd:element name="next" type="CT_String" minOccurs="0"/>
            <xsd:element name="link" type="CT_String" minOccurs="0"/>
            <xsd:element name="autoRedefine" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="hidden" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="uiPriority" type="CT_DecimalNumber" minOccurs="0"/>
            <xsd:element name="semiHidden" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="unhideWhenUsed" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="qFormat" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="locked" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personal" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personalCompose" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="personalReply" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="rsid" type="CT_LongHexNumber" minOccurs="0"/>
            <xsd:element name="pPr" type="CT_PPrGeneral" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rPr" type="CT_RPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblPr" type="CT_TblPrBase" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="trPr" type="CT_TrPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tcPr" type="CT_TcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tblStylePr" type="CT_TblStylePr" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_StyleType" use="optional"/>
        <xsd:attribute name="styleId" type="s:ST_String" use="optional"/>
        <xsd:attribute name="default" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="customStyle" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LsdException">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="locked" type="s:ST_OnOff"/>
        <xsd:attribute name="uiPriority" type="ST_DecimalNumber"/>
        <xsd:attribute name="semiHidden" type="s:ST_OnOff"/>
        <xsd:attribute name="unhideWhenUsed" type="s:ST_OnOff"/>
        <xsd:attribute name="qFormat" type="s:ST_OnOff"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_LatentStyles">
        <xsd:sequence>
            <xsd:element name="lsdException" type="CT_LsdException" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="defLockedState" type="s:ST_OnOff"/>
        <xsd:attribute name="defUIPriority" type="ST_DecimalNumber"/>
        <xsd:attribute name="defSemiHidden" type="s:ST_OnOff"/>
        <xsd:attribute name="defUnhideWhenUsed" type="s:ST_OnOff"/>
        <xsd:attribute name="defQFormat" type="s:ST_OnOff"/>
        <xsd:attribute name="count" type="ST_DecimalNumber"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Styles">
        <xsd:sequence>
            <xsd:element name="docDefaults" type="CT_DocDefaults" minOccurs="0"/>
            <xsd:element name="latentStyles" type="CT_LatentStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="CT_Style" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Panose">
        <xsd:attribute name="val" type="s:ST_Panose" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontFamily">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="decorative"/>
            <xsd:enumeration value="modern"/>
            <xsd:enumeration value="roman"/>
            <xsd:enumeration value="script"/>
            <xsd:enumeration value="swiss"/>
            <xsd:enumeration value="auto"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FontFamily">
        <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Pitch">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="fixed"/>
            <xsd:enumeration value="variable"/>
            <xsd:enumeration value="default"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Pitch">
        <xsd:attribute name="val" type="ST_Pitch" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontSig">
        <xsd:attribute name="usb0" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb1" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb2" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="usb3" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="csb0" use="required" type="ST_LongHexNumber"/>
        <xsd:attribute name="csb1" use="required" type="ST_LongHexNumber"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_FontRel">
        <xsd:complexContent>
            <xsd:extension base="CT_Rel">
                <xsd:attribute name="fontKey" type="s:ST_Guid"/>
                <xsd:attribute name="subsetted" type="s:ST_OnOff"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_Font">
        <xsd:sequence>
            <xsd:element name="altName" type="CT_String" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="panose1" type="CT_Panose" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_Charset" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_FontFamily" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notTrueType" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pitch" type="CT_Pitch" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sig" type="CT_FontSig" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedRegular" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedBold" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embedBoldItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontsList">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DivBdr">
        <xsd:sequence>
            <xsd:element name="top" type="CT_Border" minOccurs="0"/>
            <xsd:element name="left" type="CT_Border" minOccurs="0"/>
            <xsd:element name="bottom" type="CT_Border" minOccurs="0"/>
            <xsd:element name="right" type="CT_Border" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Div">
        <xsd:sequence>
            <xsd:element name="blockQuote" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="bodyDiv" type="CT_OnOff" minOccurs="0"/>
            <xsd:element name="marLeft" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marRight" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marTop" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="marBottom" type="CT_SignedTwipsMeasure"/>
            <xsd:element name="divBdr" type="CT_DivBdr" minOccurs="0"/>
            <xsd:element name="divsChild" type="CT_Divs" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_DecimalNumber" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Divs">
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="div" type="CT_Div"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_MathContent">
        <xsd:choice>
            <xsd:element ref="m:oMathPara"/>
            <xsd:element ref="m:oMath"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_BlockLevelChunkElts">
        <xsd:choice>
            <xsd:group ref="EG_ContentBlockContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_BlockLevelElts">
        <xsd:choice>
            <xsd:group ref="EG_BlockLevelChunkElts" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="altChunk" type="CT_AltChunk" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:group name="EG_RunLevelElts">
        <xsd:choice>
            <xsd:element name="proofErr" minOccurs="0" type="CT_ProofErr"/>
            <xsd:element name="permStart" minOccurs="0" type="CT_PermStart"/>
            <xsd:element name="permEnd" minOccurs="0" type="CT_Perm"/>
            <xsd:group ref="EG_RangeMarkupElements" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="ins" type="CT_RunTrackChange" minOccurs="0"/>
            <xsd:element name="del" type="CT_RunTrackChange" minOccurs="0"/>
            <xsd:element name="moveFrom" type="CT_RunTrackChange"/>
            <xsd:element name="moveTo" type="CT_RunTrackChange"/>
            <xsd:group ref="EG_MathContent" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Body">
        <xsd:sequence>
            <xsd:group ref="EG_BlockLevelElts" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="sectPr" minOccurs="0" maxOccurs="1" type="CT_SectPr"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Comments">
        <xsd:sequence>
            <xsd:element name="comment" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="comments" type="CT_Comments"/>
    <xsd:complexType name="CT_Footnotes">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="footnote" type="CT_FtnEdn" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="footnotes" type="CT_Footnotes"/>
    <xsd:complexType name="CT_Endnotes">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="endnote" type="CT_FtnEdn" minOccurs="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="endnotes" type="CT_Endnotes"/>
    <xsd:element name="hdr" type="CT_HdrFtr"/>
    <xsd:element name="ftr" type="CT_HdrFtr"/>
    <xsd:complexType name="CT_SmartTagType">
        <xsd:attribute name="namespaceuri" type="s:ST_String"/>
        <xsd:attribute name="name" type="s:ST_String"/>
        <xsd:attribute name="url" type="s:ST_String"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ThemeColor">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="dark1"/>
            <xsd:enumeration value="light1"/>
            <xsd:enumeration value="dark2"/>
            <xsd:enumeration value="light2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hyperlink"/>
            <xsd:enumeration value="followedHyperlink"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="background1"/>
            <xsd:enumeration value="text1"/>
            <xsd:enumeration value="background2"/>
            <xsd:enumeration value="text2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DocPartBehavior">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="content"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="pg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartBehavior">
        <xsd:attribute name="val" use="required" type="ST_DocPartBehavior"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartBehaviors">
        <xsd:choice>
            <xsd:element name="behavior" type="CT_DocPartBehavior" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocPartType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="autoExp"/>
            <xsd:enumeration value="toolbar"/>
            <xsd:enumeration value="speller"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="formFld"/>
            <xsd:enumeration value="bbPlcHdr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartType">
        <xsd:attribute name="val" use="required" type="ST_DocPartType"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartTypes">
        <xsd:choice>
            <xsd:element name="type" type="CT_DocPartType" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attribute name="all" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DocPartGallery">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="placeholder"/>
            <xsd:enumeration value="any"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="docParts"/>
            <xsd:enumeration value="coverPg"/>
            <xsd:enumeration value="eq"/>
            <xsd:enumeration value="ftrs"/>
            <xsd:enumeration value="hdrs"/>
            <xsd:enumeration value="pgNum"/>
            <xsd:enumeration value="tbls"/>
            <xsd:enumeration value="watermarks"/>
            <xsd:enumeration value="autoTxt"/>
            <xsd:enumeration value="txtBox"/>
            <xsd:enumeration value="pgNumT"/>
            <xsd:enumeration value="pgNumB"/>
            <xsd:enumeration value="pgNumMargins"/>
            <xsd:enumeration value="tblOfContents"/>
            <xsd:enumeration value="bib"/>
            <xsd:enumeration value="custQuickParts"/>
            <xsd:enumeration value="custCoverPg"/>
            <xsd:enumeration value="custEq"/>
            <xsd:enumeration value="custFtrs"/>
            <xsd:enumeration value="custHdrs"/>
            <xsd:enumeration value="custPgNum"/>
            <xsd:enumeration value="custTbls"/>
            <xsd:enumeration value="custWatermarks"/>
            <xsd:enumeration value="custAutoTxt"/>
            <xsd:enumeration value="custTxtBox"/>
            <xsd:enumeration value="custPgNumT"/>
            <xsd:enumeration value="custPgNumB"/>
            <xsd:enumeration value="custPgNumMargins"/>
            <xsd:enumeration value="custTblOfContents"/>
            <xsd:enumeration value="custBib"/>
            <xsd:enumeration value="custom1"/>
            <xsd:enumeration value="custom2"/>
            <xsd:enumeration value="custom3"/>
            <xsd:enumeration value="custom4"/>
            <xsd:enumeration value="custom5"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DocPartGallery">
        <xsd:attribute name="val" type="ST_DocPartGallery" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartCategory">
        <xsd:sequence>
            <xsd:element name="name" type="CT_String" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gallery" type="CT_DocPartGallery" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartName">
        <xsd:attribute name="val" type="s:ST_String" use="required"/>
        <xsd:attribute name="decorated" type="s:ST_OnOff" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPartPr">
        <xsd:all>
            <xsd:element name="name" type="CT_DocPartName" minOccurs="1"/>
            <xsd:element name="style" type="CT_String" minOccurs="0"/>
            <xsd:element name="category" type="CT_DocPartCategory" minOccurs="0"/>
            <xsd:element name="types" type="CT_DocPartTypes" minOccurs="0"/>
            <xsd:element name="behaviors" type="CT_DocPartBehaviors"
                minOccurs="0"/>
            <xsd:element name="description" type="CT_String" minOccurs="0"/>
            <xsd:element name="guid" type="CT_Guid" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="CT_DocPart">
        <xsd:sequence>
            <xsd:element name="docPartPr" type="CT_DocPartPr" minOccurs="0"/>
            <xsd:element name="docPartBody" type="CT_Body" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocParts">
        <xsd:choice>
            <xsd:element name="docPart" type="CT_DocPart" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:element name="settings" type="CT_Settings"/>
    <xsd:element name="webSettings" type="CT_WebSettings"/>
    <xsd:element name="fonts" type="CT_FontsList"/>
    <xsd:element name="numbering" type="CT_Numbering"/>
    <xsd:element name="styles" type="CT_Styles"/>
    <xsd:simpleType name="ST_CaptionPos">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="above"/>
            <xsd:enumeration value="below"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="right"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Caption">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="pos" type="ST_CaptionPos" use="optional"/>
        <xsd:attribute name="chapNum" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="heading" type="ST_DecimalNumber" use="optional"/>
        <xsd:attribute name="noLabel" type="s:ST_OnOff" use="optional"/>
        <xsd:attribute name="numFmt" type="ST_NumberFormat" use="optional"/>
        <xsd:attribute name="sep" type="ST_ChapterSep" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoCaption">
        <xsd:attribute name="name" type="s:ST_String" use="required"/>
        <xsd:attribute name="caption" type="s:ST_String" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoCaptions">
        <xsd:sequence>
            <xsd:element name="autoCaption" type="CT_AutoCaption" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Captions">
        <xsd:sequence>
            <xsd:element name="caption" type="CT_Caption" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="autoCaptions" type="CT_AutoCaptions" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DocumentBase">
        <xsd:sequence>
            <xsd:element name="background" type="CT_Background" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Document">
        <xsd:complexContent>
            <xsd:extension base="CT_DocumentBase">
                <xsd:sequence>
                    <xsd:element name="body" type="CT_Body" minOccurs="0" maxOccurs="1"/>
                </xsd:sequence>
                <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="CT_GlossaryDocument">
        <xsd:complexContent>
            <xsd:extension base="CT_DocumentBase">
                <xsd:sequence>
                    <xsd:element name="docParts" type="CT_DocParts" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="document" type="CT_Document"/>
    <xsd:element name="glossaryDocument" type="CT_GlossaryDocument"/>
</xsd:schema>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

A.2 SpreadsheetML

This schema is available in the file sml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns=http://purl.oclc.org/ooxml/spreadsheetml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:xdr="http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  targetNamespace="http://purl.oclc.org/ooxml/spreadsheetml/main" elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/spreadsheetDrawing"
      schemaLocation="dml-spreadsheetDrawing.xsd"/>
    <xsd:complexType name="CT_AutoFilter">
        <xsd:sequence>
            <xsd:element name="filterColumn" minOccurs="0" maxOccurs="unbounded"
              type="CT_FilterColumn"/>
            <xsd:element name="sortState" minOccurs="0" maxOccurs="1" type="CT_SortState"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ref" type="ST_Ref"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FilterColumn">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="filters" type="CT_Filters" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="top10" type="CT_Top10" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customFilters" type="CT_CustomFilters" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dynamicFilter" type="CT_DynamicFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colorFilter" type="CT_ColorFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="iconFilter" minOccurs="0" maxOccurs="1" type="CT_IconFilter"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="colId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="hiddenButton" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showButton" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Filters">
        <xsd:sequence>
            <xsd:element name="filter" type="CT_Filter" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="dateGroupItem" type="CT_DateGroupItem" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="blank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="calendarType" type="s:ST_CalendarType" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Filter">
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomFilters">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="customFilter" type="CT_CustomFilter" minOccurs="1" maxOccurs="2"/>
        </xsd:sequence>
        <xsd:attribute name="and" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomFilter">
        <xsd:attribute name="operator" type="ST_FilterOperator" default="equal" use="optional"/>
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Top10">
        <xsd:attribute name="top" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="percent" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="val" type="xsd:double" use="required"/>
        <xsd:attribute name="filterVal" type="xsd:double" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorFilter">
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="cellColor" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_IconFilter">
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="required"/>
        <xsd:attribute name="iconId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FilterOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
            <xsd:enumeration value="greaterThan"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DynamicFilter">
        <xsd:attribute name="type" type="ST_DynamicFilterType" use="required"/>
        <xsd:attribute name="val" type="xsd:double" use="optional"/>
        <xsd:attribute name="valIso" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="maxValIso" type="xsd:dateTime" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DynamicFilterType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="null"/>
            <xsd:enumeration value="aboveAverage"/>
            <xsd:enumeration value="belowAverage"/>
            <xsd:enumeration value="tomorrow"/>
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="nextWeek"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
            <xsd:enumeration value="nextQuarter"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="thisQuarter"/>
            <xsd:enumeration value="lastQuarter"/>
            <xsd:enumeration value="nextYear"/>
            <xsd:enumeration value="thisYear"/>
            <xsd:enumeration value="lastYear"/>
            <xsd:enumeration value="yearToDate"/>
            <xsd:enumeration value="Q1"/>
            <xsd:enumeration value="Q2"/>
            <xsd:enumeration value="Q3"/>
            <xsd:enumeration value="Q4"/>
            <xsd:enumeration value="M1"/>
            <xsd:enumeration value="M2"/>
            <xsd:enumeration value="M3"/>
            <xsd:enumeration value="M4"/>
            <xsd:enumeration value="M5"/>
            <xsd:enumeration value="M6"/>
            <xsd:enumeration value="M7"/>
            <xsd:enumeration value="M8"/>
            <xsd:enumeration value="M9"/>
            <xsd:enumeration value="M10"/>
            <xsd:enumeration value="M11"/>
            <xsd:enumeration value="M12"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_IconSetType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="3Arrows"/>
            <xsd:enumeration value="3ArrowsGray"/>
            <xsd:enumeration value="3Flags"/>
            <xsd:enumeration value="3TrafficLights1"/>
            <xsd:enumeration value="3TrafficLights2"/>
            <xsd:enumeration value="3Signs"/>
            <xsd:enumeration value="3Symbols"/>
            <xsd:enumeration value="3Symbols2"/>
            <xsd:enumeration value="4Arrows"/>
            <xsd:enumeration value="4ArrowsGray"/>
            <xsd:enumeration value="4RedToBlack"/>
            <xsd:enumeration value="4Rating"/>
            <xsd:enumeration value="4TrafficLights"/>
            <xsd:enumeration value="5Arrows"/>
            <xsd:enumeration value="5ArrowsGray"/>
            <xsd:enumeration value="5Rating"/>
            <xsd:enumeration value="5Quarters"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SortState">
        <xsd:sequence>
            <xsd:element name="sortCondition" minOccurs="0" maxOccurs="64" type="CT_SortCondition"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="columnSort" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="caseSensitive" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sortMethod" type="ST_SortMethod" use="optional" default="none"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SortCondition">
        <xsd:attribute name="descending" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sortBy" type="ST_SortBy" use="optional" default="value"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="customList" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="optional" default="3Arrows"/>
        <xsd:attribute name="iconId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SortBy">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="value"/>
            <xsd:enumeration value="cellColor"/>
            <xsd:enumeration value="fontColor"/>
            <xsd:enumeration value="icon"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SortMethod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="stroke"/>
            <xsd:enumeration value="pinYin"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DateGroupItem">
        <xsd:attribute name="year" type="xsd:unsignedShort" use="required"/>
        <xsd:attribute name="month" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="day" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="hour" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="minute" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="second" type="xsd:unsignedShort" use="optional"/>
        <xsd:attribute name="dateTimeGrouping" type="ST_DateTimeGrouping" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DateTimeGrouping">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="year"/>
            <xsd:enumeration value="month"/>
            <xsd:enumeration value="day"/>
            <xsd:enumeration value="hour"/>
            <xsd:enumeration value="minute"/>
            <xsd:enumeration value="second"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellRef">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Ref">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RefA">
        <xsd:restriction base="xsd:string"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:simpleType>
    <xsd:simpleType name="ST_Sqref">
        <xsd:list itemType="ST_Ref"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Formula">
        <xsd:restriction base="s:ST_Xstring"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_UnsignedIntHex">
        <xsd:restriction base="xsd:hexBinary">
            <xsd:length value="4"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_XStringElement">
        <xsd:attribute name="v" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Extension">
        <xsd:sequence>
            <xsd:any processContents="lax"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectAnchor">
        <xsd:sequence>
            <xsd:element ref="xdr:from" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="xdr:to" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="moveWithCells" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sizeWithCells" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:group name="EG_ExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_ExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_ExtensionList" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="calcChain" type="CT_CalcChain"/>
    <xsd:complexType name="CT_CalcChain">
        <xsd:sequence>
            <xsd:element name="c" type="CT_CalcCell" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CalcCell">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="i" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="l" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="t" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="a" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:element name="comments" type="CT_Comments"/>
    <xsd:complexType name="CT_Comments">
        <xsd:sequence>
            <xsd:element name="authors" type="CT_Authors" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="commentList" type="CT_CommentList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Authors">
        <xsd:sequence>
            <xsd:element name="author" type="s:ST_Xstring" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentList">
        <xsd:sequence>
            <xsd:element name="comment" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Comment">
        <xsd:sequence>
            <xsd:element name="text" type="CT_Rst" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="commentPr" type="CT_CommentPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="authorId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="guid" type="s:ST_Guid" use="optional"/>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="textHAlign" type="ST_TextHAlign" use="optional" default="left"/>
        <xsd:attribute name="textVAlign" type="ST_TextVAlign" use="optional" default="top"/>
        <xsd:attribute name="lockText" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="justLastX" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoScale" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TextHAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TextVAlign">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="MapInfo" type="CT_MapInfo"/>
    <xsd:complexType name="CT_MapInfo">
        <xsd:sequence>
            <xsd:element name="Schema" type="CT_Schema" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="Map" type="CT_Map" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="SelectionNamespaces" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Schema" mixed="true">
        <xsd:sequence>
            <xsd:any/>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:string" use="required"/>
        <xsd:attribute name="SchemaRef" type="xsd:string" use="optional"/>
        <xsd:attribute name="Namespace" type="xsd:string" use="optional"/>
        <xsd:attribute name="SchemaLanguage" type="xsd:token" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Map">
        <xsd:sequence>
            <xsd:element name="DataBinding" type="CT_DataBinding" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="Name" type="xsd:string" use="required"/>
        <xsd:attribute name="RootElement" type="xsd:string" use="required"/>
        <xsd:attribute name="SchemaID" type="xsd:string" use="required"/>
        <xsd:attribute name="ShowImportExportValidationErrors" type="xsd:boolean" use="required"/>
        <xsd:attribute name="AutoFit" type="xsd:boolean" use="required"/>
        <xsd:attribute name="Append" type="xsd:boolean" use="required"/>
        <xsd:attribute name="PreserveSortAFLayout" type="xsd:boolean" use="required"/>
        <xsd:attribute name="PreserveFormat" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBinding">
        <xsd:sequence>
            <xsd:any/>
        </xsd:sequence>
        <xsd:attribute name="DataBindingName" type="xsd:string" use="optional"/>
        <xsd:attribute name="FileBinding" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="ConnectionID" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="FileBindingName" type="xsd:string" use="optional"/>
        <xsd:attribute name="DataBindingLoadMode" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:element name="connections" type="CT_Connections"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_Connections">
        <xsd:sequence>
            <xsd:element name="connection" minOccurs="1" maxOccurs="unbounded" type="CT_Connection"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connection">
        <xsd:sequence>
            <xsd:element name="dbPr" minOccurs="0" maxOccurs="1" type="CT_DbPr"/>
            <xsd:element name="olapPr" minOccurs="0" maxOccurs="1" type="CT_OlapPr"/>
            <xsd:element name="webPr" minOccurs="0" maxOccurs="1" type="CT_WebPr"/>
            <xsd:element name="textPr" minOccurs="0" maxOccurs="1" type="CT_TextPr"/>
            <xsd:element name="parameters" minOccurs="0" maxOccurs="1" type="CT_Parameters"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="id" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="sourceFile" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="odcFile" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="keepAlive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="interval" use="optional" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="description" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="type" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="reconnectionMethod" use="optional" type="xsd:unsignedInt" default="1"/>
        <xsd:attribute name="refreshedVersion" use="required" type="xsd:unsignedByte"/>
        <xsd:attribute name="minRefreshableVersion" use="optional" type="xsd:unsignedByte"
          default="0"/>
        <xsd:attribute name="savePassword" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="new" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="deleted" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="onlyUseConnectionFile" use="optional" type="xsd:boolean"
          default="false"/>
        <xsd:attribute name="background" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="refreshOnLoad" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="saveData" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="credentials" use="optional" type="ST_CredMethod" default="integrated"/>
        <xsd:attribute name="singleSignOnId" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CredMethod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="integrated"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="stored"/>
            <xsd:enumeration value="prompt"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DbPr">
        <xsd:attribute name="connection" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="command" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="serverCommand" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="commandType" use="optional" type="xsd:unsignedInt" default="2"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OlapPr">
        <xsd:attribute name="local" use="optional" type="xsd:boolean" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="localConnection" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="localRefresh" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="sendLocale" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rowDrillCount" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="serverFill" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverNumberFormat" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverFont" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="serverFontColor" use="optional" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPr">
        <xsd:sequence>
            <xsd:element name="tables" minOccurs="0" maxOccurs="1" type="CT_Tables"/>
        </xsd:sequence>
        <xsd:attribute name="xml" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sourceData" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="parsePre" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="consecutive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="firstRow" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="xl97" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="textDates" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="xl2000" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="url" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="post" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="htmlTables" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="htmlFormat" use="optional" type="ST_HtmlFmt" default="none"/>
        <xsd:attribute name="editPage" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_HtmlFmt">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="rtf"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Parameters">
        <xsd:sequence>
            <xsd:element name="parameter" minOccurs="1" maxOccurs="unbounded" type="CT_Parameter"/>
        </xsd:sequence>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Parameter">
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="sqlType" use="optional" type="xsd:int" default="0"/>
        <xsd:attribute name="parameterType" use="optional" type="ST_ParameterType" default="prompt"/>
        <xsd:attribute name="refreshOnChange" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="prompt" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="boolean" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="double" use="optional" type="xsd:double"/>
        <xsd:attribute name="integer" use="optional" type="xsd:int"/>
        <xsd:attribute name="string" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="cell" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ParameterType">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="prompt"/>
            <xsd:enumeration value="value"/>
            <xsd:enumeration value="cell"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Tables">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="m" type="CT_TableMissing"/>
            <xsd:element name="s" type="CT_XStringElement"/>
            <xsd:element name="x" type="CT_Index"/>
        </xsd:choice>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableMissing"/>
    <xsd:complexType name="CT_TextPr">
        <xsd:sequence>
            <xsd:element name="textFields" minOccurs="0" maxOccurs="1" type="CT_TextFields"/>
        </xsd:sequence>
        <xsd:attribute name="prompt" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="fileType" use="optional" type="ST_FileType" default="win"/>
        <xsd:attribute name="characterSet" use="optional" type="xsd:string"/>
        <xsd:attribute name="firstRow" use="optional" type="xsd:unsignedInt" default="1"/>
        <xsd:attribute name="sourceFile" use="optional" type="s:ST_Xstring" default=""/>
        <xsd:attribute name="delimited" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="decimal" use="optional" type="s:ST_Xstring" default="."/>
        <xsd:attribute name="thousands" use="optional" type="s:ST_Xstring" default=","/>
        <xsd:attribute name="tab" use="optional" type="xsd:boolean" default="true"/>
        <xsd:attribute name="space" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="comma" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="semicolon" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="consecutive" use="optional" type="xsd:boolean" default="false"/>
        <xsd:attribute name="qualifier" use="optional" type="ST_Qualifier" default="doubleQuote"/>
        <xsd:attribute name="delimiter" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FileType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="mac"/>
            <xsd:enumeration value="win"/>
            <xsd:enumeration value="dos"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="other"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Qualifier">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="doubleQuote"/>
            <xsd:enumeration value="singleQuote"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TextFields">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="textField" minOccurs="1" maxOccurs="unbounded" type="CT_TextField"/>
        </xsd:sequence>
        <xsd:attribute name="count" use="optional" type="xsd:unsignedInt" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TextField">
        <xsd:attribute name="type" use="optional" type="ST_ExternalConnectionType" default="general"/>
        <xsd:attribute name="position" use="optional" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ExternalConnectionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="general"/>
            <xsd:enumeration value="text"/>
            <xsd:enumeration value="MDY"/>
            <xsd:enumeration value="DMY"/>
            <xsd:enumeration value="YMD"/>
            <xsd:enumeration value="MYD"/>
            <xsd:enumeration value="DYM"/>
            <xsd:enumeration value="YDM"/>
            <xsd:enumeration value="skip"/>
            <xsd:enumeration value="EMD"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="pivotCacheDefinition" type="CT_PivotCacheDefinition"/>
    <xsd:element name="pivotCacheRecords" type="CT_PivotCacheRecords"/>
    <xsd:element name="pivotTableDefinition" type="CT_pivotTableDefinition"/>
    <xsd:complexType name="CT_PivotCacheDefinition">
        <xsd:sequence>
            <xsd:element name="cacheSource" type="CT_CacheSource" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cacheFields" type="CT_CacheFields" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cacheHierarchies" minOccurs="0" type="CT_CacheHierarchies"/>
            <xsd:element name="kpis" minOccurs="0" type="CT_PCDKPIs"/>
            <xsd:element name="tupleCache" minOccurs="0" type="CT_TupleCache"/>
            <xsd:element name="calculatedItems" minOccurs="0" type="CT_CalculatedItems"/>
            <xsd:element name="calculatedMembers" type="CT_CalculatedMembers" minOccurs="0"/>
            <xsd:element name="dimensions" type="CT_Dimensions" minOccurs="0"/>
            <xsd:element name="measureGroups" type="CT_MeasureGroups" minOccurs="0"/>
            <xsd:element name="maps" type="CT_MeasureDimensionMaps" minOccurs="0"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="invalid" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveData" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="optimizeMemory" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="enableRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshedBy" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="refreshedDateIso" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="backgroundQuery" type="xsd:boolean" default="false"/>
        <xsd:attribute name="missingItemsLimit" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="createdVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="refreshedVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="minRefreshableVersion" type="xsd:unsignedByte" use="optional"
          default="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="recordCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="upgradeOnRefresh" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tupleCache" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="supportSubquery" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="supportAdvancedDrill" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheFields">
        <xsd:sequence>
            <xsd:element name="cacheField" type="CT_CacheField" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheField">
        <xsd:sequence>
            <xsd:element name="sharedItems" type="CT_SharedItems" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fieldGroup" minOccurs="0" type="CT_FieldGroup"/>
            <xsd:element name="mpMap" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="caption" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="propertyName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="serverField" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uniqueList" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="formula" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sqlType" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="hierarchy" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="level" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="databaseField" type="xsd:boolean" default="true"/>
        <xsd:attribute name="mappingCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="memberPropertyField" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheSource">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="worksheetSource" type="CT_WorksheetSource" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="consolidation" type="CT_Consolidation" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0"/>
        </xsd:choice>
        <xsd:attribute name="type" type="ST_SourceType" use="required"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" default="0" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="worksheet"/>
            <xsd:enumeration value="external"/>
            <xsd:enumeration value="consolidation"/>
            <xsd:enumeration value="scenario"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WorksheetSource">
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Consolidation">
        <xsd:sequence>
            <xsd:element name="pages" type="CT_Pages" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rangeSets" type="CT_RangeSets" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="autoPage" type="xsd:boolean" default="true" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Pages">
        <xsd:sequence>
            <xsd:element name="page" type="CT_PCDSCPage" minOccurs="1" maxOccurs="4"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDSCPage">
        <xsd:sequence>
            <xsd:element name="pageItem" type="CT_PageItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageItem">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangeSets">
        <xsd:sequence>
            <xsd:element name="rangeSet" type="CT_RangeSet" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangeSet">
        <xsd:attribute name="i1" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i2" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i3" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="i4" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SharedItems">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="n" type="CT_Number" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="b" type="CT_Boolean" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="e" type="CT_Error" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="s" type="CT_String" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="d" type="CT_DateTime" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
        <xsd:attribute name="containsSemiMixedTypes" type="xsd:boolean" use="optional"
          default="true"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="containsNonDate" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="containsDate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsString" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="containsBlank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsMixedTypes" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="containsInteger" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="minValue" type="xsd:double" use="optional"/>
        <xsd:attribute name="maxValue" type="xsd:double" use="optional"/>
        <xsd:attribute name="minDate" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="maxDate" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="longText" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Missing">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Number">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="xsd:double"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Boolean">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="v" use="required" type="xsd:boolean"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Error">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_String">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
        <xsd:attribute name="in" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="un" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DateTime">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="v" use="required" type="xsd:dateTime"/>
        <xsd:attribute name="u" type="xsd:boolean"/>
        <xsd:attribute name="f" type="xsd:boolean"/>
        <xsd:attribute name="c" type="s:ST_Xstring"/>
        <xsd:attribute name="cp" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldGroup">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:sequence>
            <xsd:element name="rangePr" minOccurs="0" type="CT_RangePr"/>
            <xsd:element name="discretePr" minOccurs="0" type="CT_DiscretePr"/>
            <xsd:element name="groupItems" minOccurs="0" type="CT_GroupItems"/>
        </xsd:sequence>
        <xsd:attribute name="par" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="base" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RangePr">
        <xsd:attribute name="autoStart" type="xsd:boolean" default="true"/>
        <xsd:attribute name="autoEnd" type="xsd:boolean" default="true"/>
        <xsd:attribute name="groupBy" type="ST_GroupBy" default="range"/>
        <xsd:attribute name="startNum" type="xsd:double"/>
        <xsd:attribute name="endNum" type="xsd:double"/>
        <xsd:attribute name="startDate" type="xsd:dateTime"/>
        <xsd:attribute name="endDate" type="xsd:dateTime"/>
        <xsd:attribute name="groupInterval" type="xsd:double" default="1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GroupBy">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="range"/>
            <xsd:enumeration value="seconds"/>
            <xsd:enumeration value="minutes"/>
            <xsd:enumeration value="hours"/>
            <xsd:enumeration value="days"/>
            <xsd:enumeration value="months"/>
            <xsd:enumeration value="quarters"/>
            <xsd:enumeration value="years"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DiscretePr">
        <xsd:sequence>
            <xsd:element name="x" maxOccurs="unbounded" type="CT_Index"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupItems">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="b" type="CT_Boolean"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
            <xsd:element name="d" type="CT_DateTime"/>
        </xsd:choice>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCacheRecords">
        <xsd:sequence>
            <xsd:element name="r" minOccurs="0" maxOccurs="unbounded" type="CT_Record"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_Record">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="b" type="CT_Boolean"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
            <xsd:element name="d" type="CT_DateTime"/>
            <xsd:element name="x" type="CT_Index"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDKPIs">
        <xsd:sequence>
            <xsd:element name="kpi" minOccurs="0" maxOccurs="unbounded" type="CT_PCDKPI"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDKPI">
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="displayFolder" type="s:ST_Xstring"/>
        <xsd:attribute name="measureGroup" type="s:ST_Xstring"/>
        <xsd:attribute name="parent" type="s:ST_Xstring"/>
        <xsd:attribute name="value" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="goal" type="s:ST_Xstring"/>
        <xsd:attribute name="status" type="s:ST_Xstring"/>
        <xsd:attribute name="trend" type="s:ST_Xstring"/>
        <xsd:attribute name="weight" type="s:ST_Xstring"/>
        <xsd:attribute name="time" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheHierarchies">
        <xsd:sequence>
            <xsd:element name="cacheHierarchy" minOccurs="0" maxOccurs="unbounded"
              type="CT_CacheHierarchy"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CacheHierarchy">
        <xsd:sequence>
            <xsd:element name="fieldsUsage" minOccurs="0" type="CT_FieldsUsage"/>
            <xsd:element name="groupLevels" minOccurs="0" type="CT_GroupLevels"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="measure" type="xsd:boolean" default="false"/>
        <xsd:attribute name="set" type="xsd:boolean" default="false"/>
        <xsd:attribute name="parentSet" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="iconSet" type="xsd:int" default="0"/>
        <xsd:attribute name="attribute" type="xsd:boolean" default="false"/>
        <xsd:attribute name="time" type="xsd:boolean" default="false"/>
        <xsd:attribute name="keyAttribute" type="xsd:boolean" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="defaultMemberUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="allUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="allCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="dimensionUniqueName" type="s:ST_Xstring"/>
        <xsd:attribute name="displayFolder" type="s:ST_Xstring"/>
        <xsd:attribute name="measureGroup" type="s:ST_Xstring"/>
        <xsd:attribute name="measures" type="xsd:boolean" default="false"/>
        <xsd:attribute name="count" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="oneField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="memberValueDatatype" use="optional" type="xsd:unsignedShort"/>
        <xsd:attribute name="unbalanced" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="unbalancedGroup" use="optional" type="xsd:boolean"/>
        <xsd:attribute name="hidden" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldsUsage">
        <xsd:sequence>
            <xsd:element name="fieldUsage" minOccurs="0" maxOccurs="unbounded" type="CT_FieldUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FieldUsage">
        <xsd:attribute name="x" use="required" type="xsd:int"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLevels">
        <xsd:sequence>
            <xsd:element name="groupLevel" maxOccurs="unbounded" type="CT_GroupLevel"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLevel">
        <xsd:sequence>
            <xsd:element name="groups" minOccurs="0" type="CT_Groups"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="user" type="xsd:boolean" default="false"/>
        <xsd:attribute name="customRollUp" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Groups">
        <xsd:sequence>
            <xsd:element name="group" maxOccurs="unbounded" type="CT_LevelGroup"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LevelGroup">
        <xsd:sequence>
            <xsd:element name="groupMembers" type="CT_GroupMembers"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueParent" type="s:ST_Xstring"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="id" type="xsd:int"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupMembers">
        <xsd:sequence>
            <xsd:element name="groupMember" maxOccurs="unbounded" type="CT_GroupMember"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupMember">
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="group" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TupleCache">
        <xsd:sequence>
            <xsd:element name="entries" minOccurs="0" type="CT_PCDSDTCEntries"/>
            <xsd:element name="sets" minOccurs="0" type="CT_Sets"/>
            <xsd:element name="queryCache" minOccurs="0" type="CT_QueryCache"/>
            <xsd:element name="serverFormats" minOccurs="0" maxOccurs="1" type="CT_ServerFormats"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ServerFormat">
        <xsd:attribute name="culture" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="format" use="optional" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ServerFormats">
        <xsd:sequence>
            <xsd:element name="serverFormat" type="CT_ServerFormat" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PCDSDTCEntries">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="m" type="CT_Missing"/>
            <xsd:element name="n" type="CT_Number"/>
            <xsd:element name="e" type="CT_Error"/>
            <xsd:element name="s" type="CT_String"/>
        </xsd:choice>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Tuples">
        <xsd:sequence>
            <xsd:element name="tpl" type="CT_Tuple" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Tuple">
        <xsd:attribute name="fld" type="xsd:unsignedInt"/>
        <xsd:attribute name="hier" type="xsd:unsignedInt"/>
        <xsd:attribute name="item" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Sets">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:sequence>
            <xsd:element name="set" maxOccurs="unbounded" type="CT_Set"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Set">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" maxOccurs="unbounded" type="CT_Tuples"/>
            <xsd:element name="sortByTuple" minOccurs="0" type="CT_Tuples"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="maxRank" use="required" type="xsd:int"/>
        <xsd:attribute name="setDefinition" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="sortType" type="ST_SortType" default="none"/>
        <xsd:attribute name="queryFailed" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SortType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="ascending"/>
            <xsd:enumeration value="descending"/>
            <xsd:enumeration value="ascendingAlpha"/>
            <xsd:enumeration value="descendingAlpha"/>
            <xsd:enumeration value="ascendingNatural"/>
            <xsd:enumeration value="descendingNatural"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_QueryCache">
        <xsd:sequence>
            <xsd:element name="query" maxOccurs="unbounded" type="CT_Query"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Query">
        <xsd:sequence>
            <xsd:element name="tpls" minOccurs="0" type="CT_Tuples"/>
        </xsd:sequence>
        <xsd:attribute name="mdx" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedItems">
        <xsd:sequence>
            <xsd:element name="calculatedItem" maxOccurs="unbounded" type="CT_CalculatedItem"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedItem">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="formula" type="s:ST_Xstring"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_CalculatedMembers">
        <xsd:sequence>
            <xsd:element name="calculatedMember" maxOccurs="unbounded" type="CT_CalculatedMember"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalculatedMember">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="mdx" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="memberName" type="s:ST_Xstring"/>
        <xsd:attribute name="hierarchy" type="s:ST_Xstring"/>
        <xsd:attribute name="parent" type="s:ST_Xstring"/>
        <xsd:attribute name="solveOrder" type="xsd:int" default="0"/>
        <xsd:attribute name="set" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_pivotTableDefinition">
        <xsd:sequence>
            <xsd:element name="location" type="CT_Location"/>
            <xsd:element name="pivotFields" type="CT_PivotFields" minOccurs="0"/>
            <xsd:element name="rowFields" type="CT_RowFields" minOccurs="0"/>
            <xsd:element name="rowItems" type="CT_rowItems" minOccurs="0"/>
            <xsd:element name="colFields" type="CT_ColFields" minOccurs="0"/>
            <xsd:element name="colItems" type="CT_colItems" minOccurs="0"/>
            <xsd:element name="pageFields" type="CT_PageFields" minOccurs="0"/>
            <xsd:element name="dataFields" type="CT_DataFields" minOccurs="0"/>
            <xsd:element name="formats" type="CT_Formats" minOccurs="0"/>
            <xsd:element name="conditionalFormats" type="CT_ConditionalFormats" minOccurs="0"/>
            <xsd:element name="chartFormats" type="CT_ChartFormats" minOccurs="0"/>
            <xsd:element name="pivotHierarchies" type="CT_PivotHierarchies" minOccurs="0"/>
            <xsd:element name="pivotTableStyleInfo" minOccurs="0" maxOccurs="1"
              type="CT_PivotTableStyle"/>
            <xsd:element name="filters" minOccurs="0" maxOccurs="1" type="CT_PivotFilters"/>
            <xsd:element name="rowHierarchiesUsage" type="CT_RowHierarchiesUsage" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="colHierarchiesUsage" type="CT_ColHierarchiesUsage" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="cacheId" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="dataOnRows" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dataPosition" type="xsd:unsignedInt" use="optional"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
        <xsd:attribute name="dataCaption" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="grandTotalCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="errorCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="showError" type="xsd:boolean" default="false"/>
        <xsd:attribute name="missingCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="showMissing" type="xsd:boolean" default="true"/>
        <xsd:attribute name="pageStyle" type="s:ST_Xstring"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="pivotTableStyle" type="s:ST_Xstring"/>
        <xsd:attribute name="vacatedStyle" type="s:ST_Xstring"/>
        <xsd:attribute name="tag" type="s:ST_Xstring"/>
        <xsd:attribute name="updatedVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="minRefreshableVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="asteriskTotals" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showItems" type="xsd:boolean" default="true"/>
        <xsd:attribute name="editData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="disableFieldList" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showCalcMbrs" type="xsd:boolean" default="true"/>
        <xsd:attribute name="visualTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showMultipleLabel" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDataDropDown" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDrill" type="xsd:boolean" default="true"/>
        <xsd:attribute name="printDrill" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showMemberPropertyTips" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showDataTips" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableWizard" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableDrill" type="xsd:boolean" default="true"/>
        <xsd:attribute name="enableFieldProperties" type="xsd:boolean" default="true"/>
        <xsd:attribute name="preserveFormatting" type="xsd:boolean" default="true"/>
        <xsd:attribute name="useAutoFormatting" type="xsd:boolean" default="false"/>
        <xsd:attribute name="pageWrap" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="pageOverThenDown" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalHiddenItems" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rowGrandTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="colGrandTotals" type="xsd:boolean" default="true"/>
        <xsd:attribute name="fieldPrintTitles" type="xsd:boolean" default="false"/>
        <xsd:attribute name="itemPrintTitles" type="xsd:boolean" default="false"/>
        <xsd:attribute name="mergeItem" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showDropZones" type="xsd:boolean" default="true"/>
        <xsd:attribute name="createdVersion" type="xsd:unsignedByte" default="0"/>
        <xsd:attribute name="indent" type="xsd:unsignedInt" default="1"/>
        <xsd:attribute name="showEmptyRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showEmptyCol" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showHeaders" type="xsd:boolean" default="true"/>
        <xsd:attribute name="compact" type="xsd:boolean" default="true"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="false"/>
        <xsd:attribute name="outlineData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="compactData" type="xsd:boolean" default="true"/>
        <xsd:attribute name="published" type="xsd:boolean" default="false"/>
        <xsd:attribute name="gridDropZones" type="xsd:boolean" default="false"/>
        <xsd:attribute name="immersive" type="xsd:boolean" default="true"/>
        <xsd:attribute name="multipleFieldFilters" type="xsd:boolean" default="true"/>
        <xsd:attribute name="chartFormat" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="rowHeaderCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="colHeaderCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="fieldListSortAscending" type="xsd:boolean" default="false"/>
        <xsd:attribute name="mdxSubqueries" type="xsd:boolean" default="false"/>
        <xsd:attribute name="customListSort" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Location">
        <xsd:attribute name="ref" use="required" type="ST_Ref"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="firstHeaderRow" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="firstDataRow" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="firstDataCol" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="rowPageCount" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="colPageCount" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFields">
        <xsd:sequence>
            <xsd:element name="pivotField" maxOccurs="unbounded" type="CT_PivotField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotField">
        <xsd:sequence>
            <xsd:element name="items" minOccurs="0" type="CT_Items"/>
            <xsd:element name="autoSortScope" minOccurs="0" type="CT_AutoSortScope"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
        <xsd:attribute name="axis" use="optional" type="ST_Axis"/>
        <xsd:attribute name="dataField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalCaption" type="s:ST_Xstring"/>
        <xsd:attribute name="showDropDowns" type="xsd:boolean" default="true"/>
        <xsd:attribute name="hiddenLevel" type="xsd:boolean" default="false"/>
        <xsd:attribute name="uniqueMemberProperty" type="s:ST_Xstring"/>
        <xsd:attribute name="compact" type="xsd:boolean" default="true"/>
        <xsd:attribute name="allDrilled" type="xsd:boolean" default="false"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="true"/>
        <xsd:attribute name="subtotalTop" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToRow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToCol" type="xsd:boolean" default="true"/>
        <xsd:attribute name="multipleItemSelectionAllowed" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dragToPage" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToData" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragOff" type="xsd:boolean" default="true"/>
        <xsd:attribute name="showAll" type="xsd:boolean" default="true"/>
        <xsd:attribute name="insertBlankRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="serverField" type="xsd:boolean" default="false"/>
        <xsd:attribute name="insertPageBreak" type="xsd:boolean" default="false"/>
        <xsd:attribute name="autoShow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="topAutoShow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="hideNewItems" type="xsd:boolean" default="false"/>
        <xsd:attribute name="measureFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="includeNewItemsInFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="itemPageCount" type="xsd:unsignedInt" default="10"/>
        <xsd:attribute name="sortType" type="ST_FieldSortType" default="manual"/>
        <xsd:attribute name="dataSourceSort" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="nonAutoSortDefault" type="xsd:boolean" default="false"/>
        <xsd:attribute name="rankBy" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="defaultSubtotal" type="xsd:boolean" default="true"/>
        <xsd:attribute name="sumSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countASubtotal" type="xsd:boolean" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="avgSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="maxSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="minSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="productSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showPropCell" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPropTip" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPropAsCaption" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="defaultAttributeDrillState" type="xsd:boolean" use="optional"
          default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AutoSortScope">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Items">
        <xsd:sequence>
            <xsd:element name="item" maxOccurs="unbounded" type="CT_Item"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Item">
        <xsd:attribute name="n" type="s:ST_Xstring"/>
        <xsd:attribute name="t" type="ST_ItemType" default="data"/>
        <xsd:attribute name="h" type="xsd:boolean" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sd" type="xsd:boolean" default="true"/>
        <xsd:attribute name="f" type="xsd:boolean" default="false"/>
        <xsd:attribute name="m" type="xsd:boolean" default="false"/>
        <xsd:attribute name="c" type="xsd:boolean" default="false"/>
        <xsd:attribute name="x" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="d" type="xsd:boolean" default="false"/>
        <xsd:attribute name="e" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageFields">
        <xsd:sequence>
            <xsd:element name="pageField" maxOccurs="unbounded" type="CT_PageField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageField">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="fld" use="required" type="xsd:int"/>
        <xsd:attribute name="item" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="hier" type="xsd:int"/>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="cap" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataFields">
        <xsd:sequence>
            <xsd:element name="dataField" maxOccurs="unbounded" type="CT_DataField"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataField">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" use="optional" type="s:ST_Xstring"/>
        <xsd:attribute name="fld" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="subtotal" type="ST_DataConsolidateFunction" default="sum"/>
        <xsd:attribute name="showDataAs" type="ST_ShowDataAs" default="normal"/>
        <xsd:attribute name="baseField" type="xsd:int" default="-1"/>
        <xsd:attribute name="baseItem" type="xsd:unsignedInt" default="1048832"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_rowItems">
        <xsd:sequence>
            <xsd:element name="i" maxOccurs="unbounded" type="CT_I"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_colItems">
        <xsd:sequence>
            <xsd:element name="i" maxOccurs="unbounded" type="CT_I"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_I">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_X"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_ItemType" default="data"/>
        <xsd:attribute name="r" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="i" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_X">
        <xsd:attribute name="v" type="xsd:int" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RowFields">
        <xsd:sequence>
            <xsd:element name="field" maxOccurs="unbounded" type="CT_Field"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColFields">
        <xsd:sequence>
            <xsd:element name="field" maxOccurs="unbounded" type="CT_Field"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Field">
        <xsd:attribute name="x" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Formats">
        <xsd:sequence>
            <xsd:element name="format" maxOccurs="unbounded" type="CT_Format"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Format">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="action" type="ST_FormatAction" default="formatting"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConditionalFormats">
        <xsd:sequence>
            <xsd:element name="conditionalFormat" maxOccurs="unbounded" type="CT_ConditionalFormat"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConditionalFormat">
        <xsd:sequence>
            <xsd:element name="pivotAreas" type="CT_PivotAreas"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="scope" type="ST_Scope" default="selection"/>
        <xsd:attribute name="type" type="ST_Type" default="none"/>
        <xsd:attribute name="priority" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotAreas">
        <xsd:sequence>
            <xsd:element name="pivotArea" minOccurs="0" maxOccurs="unbounded" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Scope">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="selection"/>
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="field"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Type">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="row"/>
            <xsd:enumeration value="column"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ChartFormats">
        <xsd:sequence>
            <xsd:element name="chartFormat" maxOccurs="unbounded" type="CT_ChartFormat"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartFormat">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="chart" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="format" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="series" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotHierarchies">
        <xsd:sequence>
            <xsd:element name="pivotHierarchy" maxOccurs="unbounded" type="CT_PivotHierarchy"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotHierarchy">
        <xsd:sequence>
            <xsd:element name="mps" minOccurs="0" type="CT_MemberProperties"/>
            <xsd:element name="members" minOccurs="0" maxOccurs="unbounded" type="CT_Members"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="outline" type="xsd:boolean" default="false"/>
        <xsd:attribute name="multipleItemSelectionAllowed" type="xsd:boolean" default="false"/>
        <xsd:attribute name="subtotalTop" type="xsd:boolean" default="false"/>
        <xsd:attribute name="showInFieldList" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToRow" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToCol" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToPage" type="xsd:boolean" default="true"/>
        <xsd:attribute name="dragToData" type="xsd:boolean" default="false"/>
        <xsd:attribute name="dragOff" type="xsd:boolean" default="true"/>
        <xsd:attribute name="includeNewItemsInFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="caption" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RowHierarchiesUsage">
        <xsd:sequence>
            <xsd:element name="rowHierarchyUsage" minOccurs="1" maxOccurs="unbounded"
              type="CT_HierarchyUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColHierarchiesUsage">
        <xsd:sequence>
            <xsd:element name="colHierarchyUsage" minOccurs="1" maxOccurs="unbounded"
              type="CT_HierarchyUsage"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:complexType name="CT_HierarchyUsage">
        <xsd:attribute name="hierarchyUsage" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MemberProperties">
        <xsd:sequence>
            <xsd:element name="mp" maxOccurs="unbounded" type="CT_MemberProperty"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MemberProperty">
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="showCell" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showTip" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAsCaption" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="nameLen" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="pPos" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="pLen" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="level" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="field" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Members">
        <xsd:sequence>
            <xsd:element name="member" maxOccurs="unbounded" type="CT_Member"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="level" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Member">
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dimensions">
        <xsd:sequence>
            <xsd:element name="dimension" minOccurs="0" maxOccurs="unbounded"
              type="CT_PivotDimension"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotDimension">
        <xsd:attribute name="measure" type="xsd:boolean" default="false"/>
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="uniqueName" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureGroups">
        <xsd:sequence>
            <xsd:element name="measureGroup" minOccurs="0" maxOccurs="unbounded"
              type="CT_MeasureGroup"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureDimensionMaps">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="map" minOccurs="0" maxOccurs="unbounded"
              type="CT_MeasureDimensionMap"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureGroup">
        <xsd:attribute name="name" use="required" type="s:ST_Xstring"/>
        <xsd:attribute name="caption" use="required" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MeasureDimensionMap">
        <xsd:attribute name="measureGroup" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="dimension" use="optional" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotTableStyle">
        <xsd:attribute name="name" type="xsd:string"/>
        <xsd:attribute name="showRowHeaders" type="xsd:boolean"/>
        <xsd:attribute name="showColHeaders" type="xsd:boolean"/>
        <xsd:attribute name="showRowStripes" type="xsd:boolean"/>
        <xsd:attribute name="showColStripes" type="xsd:boolean"/>
        <xsd:attribute name="showLastColumn" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFilters">
        <xsd:sequence>
            <xsd:element name="filter" minOccurs="0" maxOccurs="unbounded" type="CT_PivotFilter"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotFilter">
        <xsd:sequence>
            <xsd:element name="autoFilter" minOccurs="1" maxOccurs="1" type="CT_AutoFilter"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="fld" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="mpFld" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="type" use="required" type="ST_PivotFilterType"/>
        <xsd:attribute name="evalOrder" use="optional" type="xsd:int" default="0"/>
        <xsd:attribute name="id" use="required" type="xsd:unsignedInt"/>
        <xsd:attribute name="iMeasureHier" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="iMeasureFld" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="name" type="s:ST_Xstring"/>
        <xsd:attribute name="description" type="s:ST_Xstring"/>
        <xsd:attribute name="stringValue1" type="s:ST_Xstring"/>
        <xsd:attribute name="stringValue2" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_ShowDataAs">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="difference"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="percentDiff"/>
            <xsd:enumeration value="runTotal"/>
            <xsd:enumeration value="percentOfRow"/>
            <xsd:enumeration value="percentOfCol"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="percentOfTotal"/>
            <xsd:enumeration value="index"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ItemType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="countA"/>
            <xsd:enumeration value="avg"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="product"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="stdDevP"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="varP"/>
            <xsd:enumeration value="grand"/>
            <xsd:enumeration value="blank"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FormatAction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="formatting"/>
            <xsd:enumeration value="drill"/>
            <xsd:enumeration value="formula"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FieldSortType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="manual"/>
            <xsd:enumeration value="ascending"/>
            <xsd:enumeration value="descending"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PivotFilterType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unknown"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="captionEqual"/>
            <xsd:enumeration value="captionNotEqual"/>
            <xsd:enumeration value="captionBeginsWith"/>
            <xsd:enumeration value="captionNotBeginsWith"/>
            <xsd:enumeration value="captionEndsWith"/>
            <xsd:enumeration value="captionNotEndsWith"/>
            <xsd:enumeration value="captionContains"/>
            <xsd:enumeration value="captionNotContains"/>
            <xsd:enumeration value="captionGreaterThan"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="captionGreaterThanOrEqual"/>
            <xsd:enumeration value="captionLessThan"/>
            <xsd:enumeration value="captionLessThanOrEqual"/>
            <xsd:enumeration value="captionBetween"/>
            <xsd:enumeration value="captionNotBetween"/>
            <xsd:enumeration value="valueEqual"/>
            <xsd:enumeration value="valueNotEqual"/>
            <xsd:enumeration value="valueGreaterThan"/>
            <xsd:enumeration value="valueGreaterThanOrEqual"/>
            <xsd:enumeration value="valueLessThan"/>
            <xsd:enumeration value="valueLessThanOrEqual"/>
            <xsd:enumeration value="valueBetween"/>
            <xsd:enumeration value="valueNotBetween"/>
            <xsd:enumeration value="dateEqual"/>
            <xsd:enumeration value="dateNotEqual"/>
            <xsd:enumeration value="dateOlderThan"/>
            <xsd:enumeration value="dateOlderThanOrEqual"/>
            <xsd:enumeration value="dateNewerThan"/>
            <xsd:enumeration value="dateNewerThanOrEqual"/>
            <xsd:enumeration value="dateBetween"/>
            <xsd:enumeration value="dateNotBetween"/>
            <xsd:enumeration value="tomorrow"/>
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="nextWeek"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
            <xsd:enumeration value="nextQuarter"/>
            <xsd:enumeration value="thisQuarter"/>
            <xsd:enumeration value="lastQuarter"/>
            <xsd:enumeration value="nextYear"/>
            <xsd:enumeration value="thisYear"/>
            <xsd:enumeration value="lastYear"/>
            <xsd:enumeration value="yearToDate"/>
            <xsd:enumeration value="Q1"/>
            <xsd:enumeration value="Q2"/>
            <xsd:enumeration value="Q3"/>
            <xsd:enumeration value="Q4"/>
            <xsd:enumeration value="M1"/>
            <xsd:enumeration value="M2"/>
            <xsd:enumeration value="M3"/>
            <xsd:enumeration value="M4"/>
            <xsd:enumeration value="M5"/>
            <xsd:enumeration value="M6"/>
            <xsd:enumeration value="M7"/>
            <xsd:enumeration value="M8"/>
            <xsd:enumeration value="M9"/>
            <xsd:enumeration value="M10"/>
            <xsd:enumeration value="M11"/>
            <xsd:enumeration value="M12"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PivotArea">
        <xsd:sequence>
            <xsd:element name="references" minOccurs="0" type="CT_PivotAreaReferences"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" use="optional" type="xsd:int"/>
        <xsd:attribute name="type" type="ST_PivotAreaType" default="normal"/>
        <xsd:attribute name="dataOnly" type="xsd:boolean" default="true"/>
        <xsd:attribute name="labelOnly" type="xsd:boolean" default="false"/>
        <xsd:attribute name="grandRow" type="xsd:boolean" default="false"/>
        <xsd:attribute name="grandCol" type="xsd:boolean" default="false"/>
        <xsd:attribute name="cacheIndex" type="xsd:boolean" default="false"/>
        <xsd:attribute name="outline" type="xsd:boolean" default="true"/>
        <xsd:attribute name="offset" type="ST_Ref"/>
        <xsd:attribute name="collapsedLevelsAreSubtotals" type="xsd:boolean" default="false"/>
        <xsd:attribute name="axis" type="ST_Axis" use="optional"/>
        <xsd:attribute name="fieldPosition" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PivotAreaType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="data"/>
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="origin"/>
            <xsd:enumeration value="button"/>
            <xsd:enumeration value="topEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PivotAreaReferences">
        <xsd:sequence>
            <xsd:element name="reference" maxOccurs="unbounded" type="CT_PivotAreaReference"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotAreaReference">
        <xsd:sequence>
            <xsd:element name="x" minOccurs="0" maxOccurs="unbounded" type="CT_Index"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="field" use="optional" type="xsd:unsignedInt"/>
        <xsd:attribute name="count" type="xsd:unsignedInt"/>
        <xsd:attribute name="selected" type="xsd:boolean" default="true"/>
        <xsd:attribute name="byPosition" type="xsd:boolean" default="false"/>
        <xsd:attribute name="relative" type="xsd:boolean" default="false"/>
        <xsd:attribute name="defaultSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sumSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countASubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="avgSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="maxSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="minSubtotal" type="xsd:boolean" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="productSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="countSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="stdDevPSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varSubtotal" type="xsd:boolean" default="false"/>
        <xsd:attribute name="varPSubtotal" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Index">
        <xsd:attribute name="v" use="required" type="xsd:unsignedInt"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Axis">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="axisRow"/>
            <xsd:enumeration value="axisCol"/>
            <xsd:enumeration value="axisPage"/>
            <xsd:enumeration value="axisValues"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="queryTable" type="CT_QueryTable"/>
    <xsd:complexType name="CT_QueryTable">
        <xsd:sequence>
            <xsd:element name="queryTableRefresh" type="CT_QueryTableRefresh" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="headers" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rowNumbers" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="disableRefresh" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="backgroundRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="firstBackgroundRefresh" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="refreshOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="growShrinkType" type="ST_GrowShrinkType" use="optional"
          default="insertDelete"/>
        <xsd:attribute name="fillFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="removeDataOnSave" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="disableEdit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preserveFormatting" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="adjustColumnWidth" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="intermediate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="required"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableRefresh">
        <xsd:sequence>
            <xsd:element name="queryTableFields" type="CT_QueryTableFields" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="queryTableDeletedFields" type="CT_QueryTableDeletedFields"
              minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" minOccurs="0" maxOccurs="1" type="CT_SortState"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="preserveSortFilterLayout" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="fieldIdWrapped" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headersInLastRefresh" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="minimumVersion" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="nextId" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="unboundColumnsLeft" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="unboundColumnsRight" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableDeletedFields">
        <xsd:sequence>
            <xsd:element name="deletedField" type="CT_DeletedField" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DeletedField">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableFields">
        <xsd:sequence>
            <xsd:element name="queryTableField" type="CT_QueryTableField" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QueryTableField">
        <xsd:sequence minOccurs="0">
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataBound" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rowNumbers" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="fillFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clipped" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tableColumnId" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GrowShrinkType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="insertDelete"/>
            <xsd:enumeration value="insertClear"/>
            <xsd:enumeration value="overwriteClear"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="sst" type="CT_Sst"/>
    <xsd:complexType name="CT_Sst">
        <xsd:sequence>
            <xsd:element name="si" type="CT_Rst" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="uniqueCount" type="xsd:unsignedInt" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_PhoneticType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="halfwidthKatakana"/>
            <xsd:enumeration value="fullwidthKatakana"/>
            <xsd:enumeration value="Hiragana"/>
            <xsd:enumeration value="noConversion"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PhoneticAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="noControl"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PhoneticRun">
        <xsd:sequence>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="sb" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="eb" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RElt">
        <xsd:sequence>
            <xsd:element name="rPr" type="CT_RPrElt" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RPrElt">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="rFont" type="CT_FontName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="b" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="i" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="strike" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outline" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shadow" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="condense" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extend" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sz" type="CT_FontSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="u" type="CT_UnderlineProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignFontProperty" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scheme" type="CT_FontScheme" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_Rst">
        <xsd:sequence>
            <xsd:element name="t" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="r" type="CT_RElt" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rPh" type="CT_PhoneticRun" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="phoneticPr" minOccurs="0" maxOccurs="1" type="CT_PhoneticPr"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PhoneticPr">
        <xsd:attribute name="fontId" type="ST_FontId" use="required"/>
        <xsd:attribute name="type" type="ST_PhoneticType" use="optional" default="fullwidthKatakana"/>
        <xsd:attribute name="alignment" type="ST_PhoneticAlignment" use="optional" default="left"/>
    </xsd:complexType>
    <xsd:element name="headers" type="CT_RevisionHeaders"/>
    <xsd:element name="revisions" type="CT_Revisions"/>
    <xsd:complexType name="CT_RevisionHeaders">
        <xsd:sequence>
            <xsd:element name="header" type="CT_RevisionHeader" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="lastGuid" type="s:ST_Guid" use="optional"/>
        <xsd:attribute name="shared" type="xsd:boolean" default="true"/>
        <xsd:attribute name="diskRevisions" type="xsd:boolean" default="false"/>
        <xsd:attribute name="history" type="xsd:boolean" default="true"/>
        <xsd:attribute name="trackRevisions" type="xsd:boolean" default="true"/>
        <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
        <xsd:attribute name="revisionId" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="version" type="xsd:int" default="1"/>
        <xsd:attribute name="keepChangeHistory" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="protected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preserveHistory" type="xsd:unsignedInt" default="30"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Revisions">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="rrc" type="CT_RevisionRowColumn" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rm" type="CT_RevisionMove" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcv" type="CT_RevisionCustomView" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rsnm" type="CT_RevisionSheetRename" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="ris" type="CT_RevisionInsertSheet" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="raf" type="CT_RevisionAutoFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rdn" type="CT_RevisionDefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcmt" type="CT_RevisionComment" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rqt" type="CT_RevisionQueryTableField" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="rcft" type="CT_RevisionConflict" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:attributeGroup name="AG_RevData">
        <xsd:attribute name="rId" type="xsd:unsignedInt" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="ua" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ra" type="xsd:boolean" use="optional" default="false"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_RevisionHeader">
        <xsd:sequence>
            <xsd:element name="sheetIdMap" minOccurs="1" maxOccurs="1" type="CT_SheetIdMap"/>
            <xsd:element name="reviewedList" minOccurs="0" maxOccurs="1" type="CT_ReviewedRevisions"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="dateTime" type="xsd:dateTime" use="required"/>
        <xsd:attribute name="maxSheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="userName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="minRId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="maxRId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetIdMap">
        <xsd:sequence>
            <xsd:element name="sheetId" type="CT_SheetId" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetId">
        <xsd:attribute name="val" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ReviewedRevisions">
        <xsd:sequence>
            <xsd:element name="reviewed" type="CT_Reviewed" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Reviewed">
        <xsd:attribute name="rId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_UndoInfo">
        <xsd:attribute name="index" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="exp" type="ST_FormulaExpression" use="required"/>
        <xsd:attribute name="ref3D" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="array" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="v" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="nf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cs" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dr" type="ST_RefA" use="required"/>
        <xsd:attribute name="dn" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionRowColumn">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="undo" type="CT_UndoInfo" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="eol" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="action" type="ST_rwColActionType" use="required"/>
        <xsd:attribute name="edge" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionMove">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="undo" type="CT_UndoInfo" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rcc" type="CT_RevisionCellChange" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rfmt" type="CT_RevisionFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="source" type="ST_Ref" use="required"/>
        <xsd:attribute name="destination" type="ST_Ref" use="required"/>
        <xsd:attribute name="sourceSheetId" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionCustomView">
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="action" type="ST_RevisionAction" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionSheetRename">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="oldName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="newName" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionInsertSheet">
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sheetPosition" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionCellChange">
        <xsd:sequence>
            <xsd:element name="oc" type="CT_Cell" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="nc" type="CT_Cell" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="odxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ndxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="odxf" type="xsd:boolean" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="xfDxf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dxf" type="xsd:boolean" default="false"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="quotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldQuotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ph" type="xsd:boolean" default="false"/>
        <xsd:attribute name="oldPh" type="xsd:boolean" default="false"/>
        <xsd:attribute name="endOfListFormulaUpdate" type="xsd:boolean" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionFormatting">
        <xsd:sequence>
            <xsd:element name="dxf" type="CT_Dxf" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xfDxf" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="start" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="length" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionAutoFormatting">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attributeGroup ref="AG_AutoFormat"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionComment">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="cell" type="ST_CellRef" use="required"/>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="action" type="ST_RevisionAction" default="add"/>
        <xsd:attribute name="alwaysShow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="old" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenColumn" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="author" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="oldLength" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="newLength" type="xsd:unsignedInt" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionDefinedName">
        <xsd:sequence>
            <xsd:element name="formula" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oldFormula" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="localSheetId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="customView" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="function" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldFunction" type="xsd:boolean" default="false"/>
        <xsd:attribute name="functionGroupId" type="xsd:unsignedByte" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="oldFunctionGroupId" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="shortcutKey" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="oldShortcutKey" type="xsd:unsignedByte" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="oldHidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customMenu" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldCustomMenu" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="description" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldDescription" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="help" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldHelp" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="statusBar" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldStatusBar" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oldComment" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionConflict">
        <xsd:attributeGroup ref="AG_RevData"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RevisionQueryTableField">
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="fieldId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_rwColActionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="insertRow"/>
            <xsd:enumeration value="deleteRow"/>
            <xsd:enumeration value="insertCol"/>
            <xsd:enumeration value="deleteCol"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RevisionAction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="add"/>
            <xsd:enumeration value="delete"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FormulaExpression">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="ref"/>
            <xsd:enumeration value="refError"/>
            <xsd:enumeration value="area"/>
            <xsd:enumeration value="areaError"/>
            <xsd:enumeration value="computedArea"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="users" type="CT_Users"/>
    <xsd:complexType name="CT_Users">
        <xsd:sequence>
            <xsd:element name="userInfo" minOccurs="0" maxOccurs="256" type="CT_SharedUser"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SharedUser">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="id" type="xsd:int" use="required"/>
        <xsd:attribute name="dateTime" type="xsd:dateTime" use="required"/>
    </xsd:complexType>
    <xsd:element name="worksheet" type="CT_Worksheet"/>
    <xsd:element name="chartsheet" type="CT_Chartsheet"/>
    <xsd:element name="dialogsheet" type="CT_Dialogsheet"/>
    <xsd:complexType name="CT_Macrosheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_SheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dimension" type="CT_SheetDimension" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetViews" type="CT_SheetViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetFormatPr" type="CT_SheetFormatPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cols" type="CT_Cols" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="sheetData" type="CT_SheetData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataConsolidate" type="CT_DataConsolidate" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomSheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="phoneticPr" type="CT_PhoneticPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="conditionalFormatting" type="CT_ConditionalFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customProperties" type="CT_CustomProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Dialogsheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" minOccurs="0" type="CT_SheetPr"/>
            <xsd:element name="sheetViews" minOccurs="0" type="CT_SheetViews"/>
            <xsd:element name="sheetFormatPr" minOccurs="0" type="CT_SheetFormatPr"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" minOccurs="0" type="CT_CustomSheetViews"/>
            <xsd:element name="printOptions" minOccurs="0" type="CT_PrintOptions"/>
            <xsd:element name="pageMargins" minOccurs="0" type="CT_PageMargins"/>
            <xsd:element name="pageSetup" minOccurs="0" type="CT_PageSetup"/>
            <xsd:element name="headerFooter" minOccurs="0" type="CT_HeaderFooter"/>
            <xsd:element name="drawing" minOccurs="0" type="CT_Drawing"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_Controls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Worksheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_SheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dimension" type="CT_SheetDimension" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetViews" type="CT_SheetViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetFormatPr" type="CT_SheetFormatPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cols" type="CT_Cols" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="sheetData" type="CT_SheetData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sheetCalcPr" type="CT_SheetCalcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetProtection" type="CT_SheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="protectedRanges" type="CT_ProtectedRanges" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scenarios" type="CT_Scenarios" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataConsolidate" type="CT_DataConsolidate" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomSheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="mergeCells" type="CT_MergeCells" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="phoneticPr" type="CT_PhoneticPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="conditionalFormatting" type="CT_ConditionalFormatting" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="dataValidations" type="CT_DataValidations" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="hyperlinks" type="CT_Hyperlinks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customProperties" type="CT_CustomProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="cellWatches" type="CT_CellWatches" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ignoredErrors" type="CT_IgnoredErrors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTags" type="CT_SmartTags" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleObjects" type="CT_OleObjects" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_Controls" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="webPublishItems" type="CT_WebPublishItems" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="tableParts" type="CT_TableParts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetData">
        <xsd:sequence>
            <xsd:element name="row" type="CT_Row" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetCalcPr">
        <xsd:attribute name="fullCalcOnLoad" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetFormatPr">
        <xsd:attribute name="baseColWidth" type="xsd:unsignedInt" use="optional" default="8"/>
        <xsd:attribute name="defaultColWidth" type="xsd:double" use="optional"/>
        <xsd:attribute name="defaultRowHeight" type="xsd:double" use="required"/>
        <xsd:attribute name="customHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="zeroHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickTop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickBottom" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevelRow" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="outlineLevelCol" type="xsd:unsignedByte" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cols">
        <xsd:sequence>
            <xsd:element name="col" type="CT_Col" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Col">
        <xsd:attribute name="min" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="width" type="xsd:double" use="optional"/>
        <xsd:attribute name="style" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bestFit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customWidth" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="phonetic" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevel" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="collapsed" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CellSpan">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellSpans">
        <xsd:list itemType="ST_CellSpan"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Row">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:sequence>
            <xsd:element name="c" type="CT_Cell" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="spans" type="ST_CellSpans" use="optional"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="customFormat" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ht" type="xsd:double" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="customHeight" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="outlineLevel" type="xsd:unsignedByte" use="optional" default="0"/>
        <xsd:attribute name="collapsed" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickTop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="thickBot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ph" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cell">
        <xsd:sequence>
            <xsd:element name="f" type="CT_CellFormula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="is" type="CT_Rst" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="t" type="ST_CellType" use="optional" default="n"/>
        <xsd:attribute name="cm" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="vm" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="ph" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CellType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="d"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="str"/>
            <xsd:enumeration value="inlineStr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellFormulaType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="array"/>
            <xsd:enumeration value="dataTable"/>
            <xsd:enumeration value="shared"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SheetPr">
        <xsd:sequence>
            <xsd:element name="tabColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="outlinePr" type="CT_OutlinePr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetUpPr" type="CT_PageSetUpPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="syncHorizontal" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="syncVertical" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="syncRef" type="ST_Ref" use="optional"/>
        <xsd:attribute name="transitionEvaluation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="transitionEntry" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
        <xsd:attribute name="filterMode" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="enableFormatConditionsCalculation" type="xsd:boolean" use="optional"
          default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetDimension">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetViews">
        <xsd:sequence>
            <xsd:element name="sheetView" type="CT_SheetView" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetView">
        <xsd:sequence>
            <xsd:element name="pane" type="CT_Pane" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="selection" type="CT_Selection" minOccurs="0" maxOccurs="4"/>
            <xsd:element name="pivotSelection" type="CT_PivotSelection" minOccurs="0" maxOccurs="4"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="windowProtection" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGridLines" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showRowColHeaders" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showZeros" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="rightToLeft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="tabSelected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showRuler" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showOutlineSymbols" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultGridColor" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showWhiteSpace" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="view" type="ST_SheetViewType" use="optional" default="normal"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="colorId" type="xsd:unsignedInt" use="optional" default="64"/>
        <xsd:attribute name="zoomScale" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="zoomScaleNormal" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="zoomScaleSheetLayoutView" type="xsd:unsignedInt" use="optional"
          default="0"/>
        <xsd:attribute name="zoomScalePageLayoutView" type="xsd:unsignedInt" use="optional"
          default="0"/>
        <xsd:attribute name="workbookViewId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Pane">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="xSplit" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="ySplit" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="activePane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="state" type="ST_PaneState" use="optional" default="split"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotSelection">
        <xsd:sequence>
            <xsd:element name="pivotArea" type="CT_PivotArea"/>
        </xsd:sequence>
        <xsd:attribute name="pane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="showHeader" type="xsd:boolean" default="false"/>
        <xsd:attribute name="label" type="xsd:boolean" default="false"/>
        <xsd:attribute name="data" type="xsd:boolean" default="false"/>
        <xsd:attribute name="extendable" type="xsd:boolean" default="false"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="axis" type="ST_Axis" use="optional"/>
        <xsd:attribute name="dimension" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="start" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="min" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="activeRow" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="activeCol" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="previousRow" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="previousCol" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute name="click" type="xsd:unsignedInt" default="0"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Selection">
        <xsd:attribute name="pane" type="ST_Pane" use="optional" default="topLeft"/>
        <xsd:attribute name="activeCell" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="activeCellId" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="optional" default="A1"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Pane">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="bottomRight"/>
            <xsd:enumeration value="topRight"/>
            <xsd:enumeration value="bottomLeft"/>
            <xsd:enumeration value="topLeft"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PageBreak">
        <xsd:sequence>
            <xsd:element name="brk" type="CT_Break" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="manualBreakCount" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Break">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="min" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="max" type="xsd:unsignedInt" use="optional" default="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="man" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pt" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SheetViewType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="pageBreakPreview"/>
            <xsd:enumeration value="pageLayout"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OutlinePr">
        <xsd:attribute name="applyStyles" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="summaryBelow" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="summaryRight" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showOutlineSymbols" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageSetUpPr">
        <xsd:attribute name="autoPageBreaks" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fitToPage" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataConsolidate">
        <xsd:sequence>
            <xsd:element name="dataRefs" type="CT_DataRefs" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="function" type="ST_DataConsolidateFunction" use="optional"
          default="sum"/>
        <xsd:attribute name="startLabels" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="topLabels" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="link" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DataConsolidateFunction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="average"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="countNums"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="product"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="stdDevp"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="varp"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DataRefs">
        <xsd:sequence>
            <xsd:element name="dataRef" type="CT_DataRef" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataRef">
        <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheet" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MergeCells">
        <xsd:sequence>
            <xsd:element name="mergeCell" type="CT_MergeCell" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MergeCell">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTags">
        <xsd:sequence>
            <xsd:element name="cellSmartTags" type="CT_CellSmartTags" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTags">
        <xsd:sequence>
            <xsd:element name="cellSmartTag" type="CT_CellSmartTag" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTag">
        <xsd:sequence>
            <xsd:element name="cellSmartTagPr" minOccurs="0" maxOccurs="unbounded"
              type="CT_CellSmartTagPr"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="deleted" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xmlBased" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellSmartTagPr">
        <xsd:attribute name="key" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Drawing">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DrawingHF">
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="lho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lhe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lhf" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="che" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="chf" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rho" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rhe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rhf" type="xsd:unsignedInt" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="lfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="lff" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="cff" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rfo" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rfe" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rff" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomSheetViews">
        <xsd:sequence>
            <xsd:element name="customSheetView" minOccurs="1" maxOccurs="unbounded"
              type="CT_CustomSheetView"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomSheetView">
        <xsd:sequence>
            <xsd:element name="pane" type="CT_Pane" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="selection" type="CT_Selection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rowBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colBreaks" type="CT_PageBreak" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="printOptions" type="CT_PrintOptions" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_PageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" default="100"/>
        <xsd:attribute name="colorId" type="xsd:unsignedInt" default="64"/>
        <xsd:attribute name="showPageBreaks" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGridLines" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showRowCol" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="outlineSymbols" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="zeroValues" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="fitToPage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="printArea" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="filter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAutoFilter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenRows" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hiddenColumns" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="state" type="ST_SheetState" default="visible"/>
        <xsd:attribute name="filterUnique" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="view" type="ST_SheetViewType" default="normal"/>
        <xsd:attribute name="showRuler" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="topLeftCell" type="ST_CellRef" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataValidations">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="dataValidation" type="CT_DataValidation" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="disablePrompts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xWindow" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="yWindow" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DataValidation">
        <xsd:sequence>
            <xsd:element name="formula1" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="formula2" type="ST_Formula" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_DataValidationType" use="optional" default="none"/>
        <xsd:attribute name="errorStyle" type="ST_DataValidationErrorStyle" use="optional"
          default="stop"/>
        <xsd:attribute name="imeMode" type="ST_DataValidationImeMode" use="optional"
          default="noControl"/>
        <xsd:attribute name="operator" type="ST_DataValidationOperator" use="optional"
          default="between"/>
        <xsd:attribute name="allowBlank" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showDropDown" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showInputMessage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showErrorMessage" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="errorTitle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="error" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="promptTitle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="prompt" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DataValidationType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="whole"/>
            <xsd:enumeration value="decimal"/>
            <xsd:enumeration value="list"/>
            <xsd:enumeration value="date"/>
            <xsd:enumeration value="time"/>
            <xsd:enumeration value="textLength"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="between"/>
            <xsd:enumeration value="notBetween"/>
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="greaterThan"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
        </xsd:restriction>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationErrorStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="stop"/>
            <xsd:enumeration value="warning"/>
            <xsd:enumeration value="information"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DataValidationImeMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="noControl"/>
            <xsd:enumeration value="off"/>
            <xsd:enumeration value="on"/>
            <xsd:enumeration value="disabled"/>
            <xsd:enumeration value="hiragana"/>
            <xsd:enumeration value="fullKatakana"/>
            <xsd:enumeration value="halfKatakana"/>
            <xsd:enumeration value="fullAlpha"/>
            <xsd:enumeration value="halfAlpha"/>
            <xsd:enumeration value="fullHangul"/>
            <xsd:enumeration value="halfHangul"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CfType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="expression"/>
            <xsd:enumeration value="cellIs"/>
            <xsd:enumeration value="colorScale"/>
            <xsd:enumeration value="dataBar"/>
            <xsd:enumeration value="iconSet"/>
            <xsd:enumeration value="top10"/>
            <xsd:enumeration value="uniqueValues"/>
            <xsd:enumeration value="duplicateValues"/>
            <xsd:enumeration value="containsText"/>
            <xsd:enumeration value="notContainsText"/>
            <xsd:enumeration value="beginsWith"/>
            <xsd:enumeration value="endsWith"/>
            <xsd:enumeration value="containsBlanks"/>
            <xsd:enumeration value="notContainsBlanks"/>
            <xsd:enumeration value="containsErrors"/>
            <xsd:enumeration value="notContainsErrors"/>
            <xsd:enumeration value="timePeriod"/>
            <xsd:enumeration value="aboveAverage"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TimePeriod">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="today"/>
            <xsd:enumeration value="yesterday"/>
            <xsd:enumeration value="tomorrow"/>
            <xsd:enumeration value="last7Days"/>
            <xsd:enumeration value="thisMonth"/>
            <xsd:enumeration value="lastMonth"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="nextMonth"/>
            <xsd:enumeration value="thisWeek"/>
            <xsd:enumeration value="lastWeek"/>
            <xsd:enumeration value="nextWeek"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ConditionalFormattingOperator">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="lessThan"/>
            <xsd:enumeration value="lessThanOrEqual"/>
            <xsd:enumeration value="equal"/>
            <xsd:enumeration value="notEqual"/>
            <xsd:enumeration value="greaterThanOrEqual"/>
            <xsd:enumeration value="greaterThan"/>
            <xsd:enumeration value="between"/>
            <xsd:enumeration value="notBetween"/>
            <xsd:enumeration value="containsText"/>
            <xsd:enumeration value="notContains"/>
            <xsd:enumeration value="beginsWith"/>
            <xsd:enumeration value="endsWith"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CfvoType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="num"/>
            <xsd:enumeration value="percent"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="formula"/>
            <xsd:enumeration value="percentile"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ConditionalFormatting">
        <xsd:sequence>
            <xsd:element name="cfRule" type="CT_CfRule" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="pivot" type="xsd:boolean" default="false"/>
        <xsd:attribute name="sqref" type="ST_Sqref"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CfRule">
        <xsd:sequence>
            <xsd:element name="formula" type="ST_Formula" minOccurs="0" maxOccurs="3"/>
            <xsd:element name="colorScale" type="CT_ColorScale" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dataBar" type="CT_DataBar" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="iconSet" type="CT_IconSet" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_CfType"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="priority" type="xsd:int" use="required"/>
        <xsd:attribute name="stopIfTrue" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="aboveAverage" type="xsd:boolean" use="optional" default="true"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="percent" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="bottom" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="operator" type="ST_ConditionalFormattingOperator" use="optional"/>
        <xsd:attribute name="text" type="xsd:string" use="optional"/>
        <xsd:attribute name="timePeriod" type="ST_TimePeriod" use="optional"/>
        <xsd:attribute name="rank" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="stdDev" type="xsd:int" use="optional"/>
        <xsd:attribute name="equalAverage" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlinks">
        <xsd:sequence>
            <xsd:element name="hyperlink" type="CT_Hyperlink" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="location" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="tooltip" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="display" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellFormula">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="t" type="ST_CellFormulaType" use="optional" default="normal"/>
                <xsd:attribute name="aca" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="ref" type="ST_Ref" use="optional"/>
                <xsd:attribute name="dt2D" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="dtr" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="del1" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="del2" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="r1" type="ST_CellRef" use="optional"/>
                <xsd:attribute name="r2" type="ST_CellRef" use="optional"/>
                <xsd:attribute name="ca" type="xsd:boolean" use="optional" default="false"/>
                <xsd:attribute name="si" type="xsd:unsignedInt" use="optional"/>
                <xsd:attribute name="bx" type="xsd:boolean" use="optional" default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorScale">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="unbounded"/>
            <xsd:element name="color" type="CT_Color" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DataBar">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="2"/>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="minLength" type="xsd:unsignedInt" use="optional" default="10"/>
        <xsd:attribute name="maxLength" type="xsd:unsignedInt" use="optional" default="90"/>
        <xsd:attribute name="showValue" type="xsd:boolean" use="optional" default="true"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_IconSet">
        <xsd:sequence>
            <xsd:element name="cfvo" type="CT_Cfvo" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="iconSet" type="ST_IconSetType" use="optional" default="3TrafficLights1"/>
        <xsd:attribute name="showValue" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="percent" type="xsd:boolean" default="true"/>
        <xsd:attribute name="reverse" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Cfvo">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_CfvoType" use="required"/>
        <xsd:attribute name="val" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="gte" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageMargins">
        <xsd:attribute name="left" type="xsd:double" use="required"/>
        <xsd:attribute name="right" type="xsd:double" use="required"/>
        <xsd:attribute name="top" type="xsd:double" use="required"/>
        <xsd:attribute name="bottom" type="xsd:double" use="required"/>
        <xsd:attribute name="header" type="xsd:double" use="required"/>
        <xsd:attribute name="footer" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PrintOptions">
        <xsd:attribute name="horizontalCentered" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="verticalCentered" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headings" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="gridLines" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="gridLinesSet" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PageSetup">
        <xsd:attribute name="paperSize" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="paperHeight" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="paperWidth" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="firstPageNumber" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="fitToWidth" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="fitToHeight" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="pageOrder" type="ST_PageOrder" use="optional" default="downThenOver"/>
        <xsd:attribute name="orientation" type="ST_Orientation" use="optional" default="default"/>
        <xsd:attribute name="usePrinterDefaults" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="blackAndWhite" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="draft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cellComments" type="ST_CellComments" use="optional" default="none"/>
        <xsd:attribute name="useFirstPageNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="errors" type="ST_PrintError" use="optional" default="displayed"/>
        <xsd:attribute name="horizontalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="verticalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="copies" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute ref="r:id" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_PageOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="downThenOver"/>
            <xsd:enumeration value="overThenDown"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Orientation">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="default"/>
            <xsd:enumeration value="portrait"/>
            <xsd:enumeration value="landscape"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellComments">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="asDisplayed"/>
            <xsd:enumeration value="atEnd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_HeaderFooter">
        <xsd:sequence>
            <xsd:element name="oddHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oddFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="evenFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstHeader" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="firstFooter" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="differentOddEven" type="xsd:boolean" default="false"/>
        <xsd:attribute name="differentFirst" type="xsd:boolean" default="false"/>
        <xsd:attribute name="scaleWithDoc" type="xsd:boolean" default="true"/>
        <xsd:attribute name="alignWithMargins" type="xsd:boolean" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PrintError">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="displayed"/>
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="dash"/>
            <xsd:enumeration value="NA"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Scenarios">
        <xsd:sequence>
            <xsd:element name="scenario" type="CT_Scenario" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="current" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="show" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="sqref" type="ST_Sqref" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetProtection">
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="sheet" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="objects" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="scenarios" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formatCells" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="formatColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="formatRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="insertHyperlinks" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="deleteColumns" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="deleteRows" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="selectLockedCells" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="sort" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoFilter" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="pivotTables" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="selectUnlockedCells" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ProtectedRanges">
        <xsd:sequence>
            <xsd:element name="protectedRange" type="CT_ProtectedRange" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ProtectedRange">
        <xsd:sequence>
            <xsd:element name="securityDescriptor" type="xsd:string" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Scenario">
        <xsd:sequence>
            <xsd:element name="inputCells" type="CT_InputCells" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="user" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InputCells">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="deleted" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="undone" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellWatches">
        <xsd:sequence>
            <xsd:element name="cellWatch" type="CT_CellWatch" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellWatch">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Chartsheet">
        <xsd:sequence>
            <xsd:element name="sheetPr" type="CT_ChartsheetPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheetViews" type="CT_ChartsheetViews" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sheetProtection" type="CT_ChartsheetProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="customSheetViews" type="CT_CustomChartsheetViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="pageMargins" minOccurs="0" type="CT_PageMargins"/>
            <xsd:element name="pageSetup" type="CT_CsPageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" minOccurs="0" type="CT_HeaderFooter"/>
            <xsd:element name="drawing" type="CT_Drawing" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="drawingHF" type="CT_DrawingHF" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="picture" type="CT_SheetBackgroundPicture" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="webPublishItems" type="CT_WebPublishItems" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetPr">
        <xsd:sequence>
            <xsd:element name="tabColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetViews">
        <xsd:sequence>
            <xsd:element name="sheetView" type="CT_ChartsheetView" minOccurs="1"
              maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ChartsheetView">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="tabSelected" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="zoomScale" type="xsd:unsignedInt" default="100" use="optional"/>
        <xsd:attribute name="workbookViewId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="zoomToFit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_ChartsheetProtection">
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="content" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="objects" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CsPageSetup">
        <xsd:attribute name="paperSize" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="paperHeight" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="paperWidth" type="s:ST_PositiveUniversalMeasure" use="optional"/>
        <xsd:attribute name="firstPageNumber" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="orientation" type="ST_Orientation" use="optional" default="default"/>
        <xsd:attribute name="usePrinterDefaults" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="blackAndWhite" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="draft" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="useFirstPageNumber" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="horizontalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="verticalDpi" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="copies" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomChartsheetViews">
        <xsd:sequence>
            <xsd:element name="customSheetView" minOccurs="0" maxOccurs="unbounded"
              type="CT_CustomChartsheetView"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomChartsheetView">
        <xsd:sequence>
            <xsd:element name="pageMargins" type="CT_PageMargins" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pageSetup" type="CT_CsPageSetup" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="headerFooter" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="scale" type="xsd:unsignedInt" default="100"/>
        <xsd:attribute name="state" type="ST_SheetState" default="visible"/>
        <xsd:attribute name="zoomToFit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomProperties">
        <xsd:sequence>
            <xsd:element name="customPr" type="CT_CustomProperty" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomProperty">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObjects">
        <xsd:sequence>
            <xsd:element name="oleObject" type="CT_OleObject" minOccurs="1" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObject">
        <xsd:sequence>
            <xsd:element name="objectPr" type="CT_ObjectPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="progId" type="xsd:string" use="optional"/>
        <xsd:attribute name="dvAspect" type="ST_DvAspect" use="optional" default="DVASPECT_CONTENT"/>
        <xsd:attribute name="link" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="oleUpdate" type="ST_OleUpdate" use="optional"/>
        <xsd:attribute name="autoLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ObjectPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uiObject" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoPict" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="macro" type="ST_Formula" use="optional"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dde" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DvAspect">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="DVASPECT_CONTENT"/>
            <xsd:enumeration value="DVASPECT_ICON"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_OleUpdate">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="OLEUPDATE_ALWAYS"/>
            <xsd:enumeration value="OLEUPDATE_ONCALL"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WebPublishItems">
        <xsd:sequence>
            <xsd:element name="webPublishItem" type="CT_WebPublishItem" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishItem">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="divId" type="s:ST_Xstring" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="sourceType" type="ST_WebSourceType" use="required"/>
        <xsd:attribute name="sourceRef" type="ST_Ref" use="optional"/>
        <xsd:attribute name="sourceObject" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="destinationFile" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="title" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="autoRepublish" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Controls">
        <xsd:sequence>
            <xsd:element name="control" type="CT_Control" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Control">
        <xsd:sequence>
            <xsd:element name="controlPr" type="CT_ControlPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="shapeId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ControlPr">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_ObjectAnchor" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="locked" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="defaultSize" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="print" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="disabled" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="recalcAlways" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="uiObject" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoFill" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoLine" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="autoPict" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="macro" type="ST_Formula" use="optional"/>
        <xsd:attribute name="altText" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="linkedCell" type="ST_Formula" use="optional"/>
        <xsd:attribute name="listFillRange" type="ST_Formula" use="optional"/>
        <xsd:attribute name="cf" type="s:ST_Xstring" use="optional" default="pict"/>
        <xsd:attribute ref="r:id" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_WebSourceType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="sheet"/>
            <xsd:enumeration value="printArea"/>
            <xsd:enumeration value="autoFilter"/>
            <xsd:enumeration value="range"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="pivotTable"/>
            <xsd:enumeration value="query"/>
            <xsd:enumeration value="label"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_IgnoredErrors">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="ignoredError" type="CT_IgnoredError" minOccurs="1"
              maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_IgnoredError">
        <xsd:attribute name="sqref" type="ST_Sqref" use="required"/>
        <xsd:attribute name="evalError" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="twoDigitTextYear" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="numberStoredAsText" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formula" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="formulaRange" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="unlockedFormula" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="emptyCellReference" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="listDataValidation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="calculatedColumn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PaneState">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="split"/>
            <xsd:enumeration value="frozen"/>
            <xsd:enumeration value="frozenSplit"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TableParts">
        <xsd:sequence>
            <xsd:element name="tablePart" type="CT_TablePart" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TablePart">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:element name="metadata" type="CT_Metadata"/>
    <xsd:complexType name="CT_Metadata">
        <xsd:sequence>
            <xsd:element name="metadataTypes" type="CT_MetadataTypes" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="metadataStrings" type="CT_MetadataStrings" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="mdxMetadata" type="CT_MdxMetadata" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="futureMetadata" type="CT_FutureMetadata" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="cellMetadata" type="CT_MetadataBlocks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="valueMetadata" type="CT_MetadataBlocks" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataTypes">
        <xsd:sequence>
            <xsd:element name="metadataType" type="CT_MetadataType" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataType">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="minSupportedVersion" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="ghostRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="ghostCol" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="edit" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="delete" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="copy" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteAll" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteFormulas" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteValues" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteComments" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteDataValidation" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteBorders" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteColWidths" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pasteNumberFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="merge" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="splitFirst" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="splitAll" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="rowColShift" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearAll" type="xsd:boolean" default="false"/>
        <xsd:attribute name="clearFormats" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearContents" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="clearComments" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="assign" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="coerce" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="adjust" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="cellMeta" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataBlocks">
        <xsd:sequence>
            <xsd:element name="bk" type="CT_MetadataBlock" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataBlock">
        <xsd:sequence>
            <xsd:element name="rc" type="CT_MetadataRecord" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataRecord">
        <xsd:attribute name="t" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="v" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FutureMetadata">
        <xsd:sequence>
            <xsd:element name="bk" type="CT_FutureMetadataBlock" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FutureMetadataBlock">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" maxOccurs="1" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MdxMetadata">
        <xsd:sequence>
            <xsd:element name="mdx" type="CT_Mdx" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Mdx">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="t" type="CT_MdxTuple"/>
            <xsd:element name="ms" type="CT_MdxSet"/>
            <xsd:element name="p" type="CT_MdxMemeberProp"/>
            <xsd:element name="k" type="CT_MdxKPI"/>
        </xsd:choice>
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="f" type="ST_MdxFunctionType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxFunctionType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="m"/>
            <xsd:enumeration value="v"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="c"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="k"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MdxTuple">
        <xsd:sequence>
            <xsd:element name="n" type="CT_MetadataStringIndex" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="ct" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="si" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="fi" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="bc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="fc" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="i" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="u" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="st" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="b" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MdxSet">
        <xsd:sequence>
            <xsd:element name="n" type="CT_MetadataStringIndex" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="ns" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="c" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="o" type="ST_MdxSetOrder" use="optional" default="u"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxSetOrder">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="u"/>
            <xsd:enumeration value="a"/>
            <xsd:enumeration value="d"/>
            <xsd:enumeration value="aa"/>
            <xsd:enumeration value="ad"/>
            <xsd:enumeration value="na"/>
            <xsd:enumeration value="nd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MdxMemeberProp">
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="np" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MdxKPI">
        <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="np" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="p" type="ST_MdxKPIProperty" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_MdxKPIProperty">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="v"/>
            <xsd:enumeration value="g"/>
            <xsd:enumeration value="s"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="w"/>
            <xsd:enumeration value="m"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_MetadataStringIndex">
        <xsd:attribute name="x" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="s" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_MetadataStrings">
        <xsd:sequence>
            <xsd:element name="s" type="CT_XStringElement" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:element name="singleXmlCells" type="CT_SingleXmlCells"/>
    <xsd:complexType name="CT_SingleXmlCells">
        <xsd:sequence>
            <xsd:element name="singleXmlCell" type="CT_SingleXmlCell" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SingleXmlCell">
        <xsd:sequence>
            <xsd:element name="xmlCellPr" type="CT_XmlCellPr" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XmlCellPr">
        <xsd:sequence>
            <xsd:element name="xmlPr" type="CT_XmlPr" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uniqueName" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_XmlPr">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mapId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xpath" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="xmlDataType" type="ST_XmlDataType" use="required"/>
    </xsd:complexType>
    <xsd:element name="styleSheet" type="CT_Stylesheet"/>
    <xsd:complexType name="CT_Stylesheet">
        <xsd:sequence>
            <xsd:element name="numFmts" type="CT_NumFmts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fonts" type="CT_Fonts" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fills" type="CT_Fills" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="borders" type="CT_Borders" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellStyleXfs" type="CT_CellStyleXfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellXfs" type="CT_CellXfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="cellStyles" type="CT_CellStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="dxfs" type="CT_Dxfs" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tableStyles" type="CT_TableStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="colors" type="CT_Colors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CellAlignment">
        <xsd:attribute name="horizontal" type="ST_HorizontalAlignment" use="optional"/>
        <xsd:attribute name="vertical" type="ST_VerticalAlignment" use="optional"/>
        <xsd:attribute name="textRotation" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="wrapText" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="indent" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="relativeIndent" type="xsd:int" use="optional"/>
        <xsd:attribute name="justifyLastLine" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="shrinkToFit" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="readingOrder" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BorderStyle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="thin"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="medium"/>
            <xsd:enumeration value="dashed"/>
            <xsd:enumeration value="dotted"/>
            <xsd:enumeration value="thick"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="hair"/>
            <xsd:enumeration value="mediumDashed"/>
            <xsd:enumeration value="dashDot"/>
            <xsd:enumeration value="mediumDashDot"/>
            <xsd:enumeration value="dashDotDot"/>
            <xsd:enumeration value="mediumDashDotDot"/>
            <xsd:enumeration value="slantDashDot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Borders">
        <xsd:sequence>
            <xsd:element name="border" type="CT_Border" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Border">
        <xsd:sequence>
            <xsd:element name="start" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="end" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="top" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bottom" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="diagonal" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertical" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="horizontal" type="CT_BorderPr" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="diagonalUp" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="diagonalDown" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="outline" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BorderPr">
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="style" type="ST_BorderStyle" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellProtection">
        <xsd:attribute name="locked" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fonts">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fills">
        <xsd:sequence>
            <xsd:element name="fill" type="CT_Fill" minOccurs="0" maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Fill">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="patternFill" type="CT_PatternFill" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="gradientFill" type="CT_GradientFill" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_PatternFill">
        <xsd:sequence>
            <xsd:element name="fgColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bgColor" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="patternType" type="ST_PatternType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Color">
        <xsd:attribute name="auto" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="indexed" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="rgb" type="ST_UnsignedIntHex" use="optional"/>
        <xsd:attribute name="theme" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="tint" type="xsd:double" use="optional" default="0.0"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PatternType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="solid"/>
            <xsd:enumeration value="mediumGray"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="darkHorizontal"/>
            <xsd:enumeration value="darkVertical"/>
            <xsd:enumeration value="darkDown"/>
            <xsd:enumeration value="darkUp"/>
            <xsd:enumeration value="darkGrid"/>
            <xsd:enumeration value="darkTrellis"/>
            <xsd:enumeration value="lightHorizontal"/>
            <xsd:enumeration value="lightVertical"/>
            <xsd:enumeration value="lightDown"/>
            <xsd:enumeration value="lightUp"/>
            <xsd:enumeration value="lightGrid"/>
            <xsd:enumeration value="lightTrellis"/>
            <xsd:enumeration value="gray125"/>
            <xsd:enumeration value="gray0625"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_GradientFill">
        <xsd:sequence>
            <xsd:element name="stop" type="CT_GradientStop" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_GradientType" use="optional" default="linear"/>
        <xsd:attribute name="degree" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="left" type="xsd:double" use="optional" default="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="right" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="top" type="xsd:double" use="optional" default="0"/>
        <xsd:attribute name="bottom" type="xsd:double" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GradientStop">
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="position" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_GradientType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="linear"/>
            <xsd:enumeration value="path"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_HorizontalAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="general"/>
            <xsd:enumeration value="left"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="right"/>
            <xsd:enumeration value="fill"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="centerContinuous"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VerticalAlignment">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="top"/>
            <xsd:enumeration value="center"/>
            <xsd:enumeration value="bottom"/>
            <xsd:enumeration value="justify"/>
            <xsd:enumeration value="distributed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NumFmts">
        <xsd:sequence>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NumFmt">
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="required"/>
        <xsd:attribute name="formatCode" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyleXfs">
        <xsd:sequence>
            <xsd:element name="xf" type="CT_Xf" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_CellXfs">
        <xsd:sequence>
            <xsd:element name="xf" type="CT_Xf" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Xf">
        <xsd:sequence>
            <xsd:element name="alignment" type="CT_CellAlignment" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="protection" type="CT_CellProtection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="numFmtId" type="ST_NumFmtId" use="optional"/>
        <xsd:attribute name="fontId" type="ST_FontId" use="optional"/>
        <xsd:attribute name="fillId" type="ST_FillId" use="optional"/>
        <xsd:attribute name="borderId" type="ST_BorderId" use="optional"/>
        <xsd:attribute name="xfId" type="ST_CellStyleXfId" use="optional"/>
        <xsd:attribute name="quotePrefix" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="pivotButton" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="applyNumberFormat" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyFont" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyFill" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyBorder" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyAlignment" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="applyProtection" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyles">
        <xsd:sequence>
            <xsd:element name="cellStyle" type="CT_CellStyle" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CellStyle">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="xfId" type="ST_CellStyleXfId" use="required"/>
        <xsd:attribute name="builtinId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="iLevel" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="customBuiltin" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dxfs">
        <xsd:sequence>
            <xsd:element name="dxf" type="CT_Dxf" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Dxf">
        <xsd:sequence>
            <xsd:element name="font" type="CT_Font" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="numFmt" type="CT_NumFmt" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="fill" type="CT_Fill" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="alignment" type="CT_CellAlignment" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="border" type="CT_Border" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="protection" type="CT_CellProtection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_NumFmtId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FontId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FillId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_BorderId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CellStyleXfId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DxfId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Colors">
        <xsd:sequence>
            <xsd:element name="indexedColors" type="CT_IndexedColors" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="mruColors" type="CT_MRUColors" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_IndexedColors">
        <xsd:sequence>
            <xsd:element name="rgbColor" type="CT_RgbColor" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_MRUColors">
        <xsd:sequence>
            <xsd:element name="color" type="CT_Color" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_RgbColor">
        <xsd:attribute name="rgb" type="ST_UnsignedIntHex" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyles">
        <xsd:sequence>
            <xsd:element name="tableStyle" type="CT_TableStyle" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="defaultTableStyle" type="xsd:string" use="optional"/>
        <xsd:attribute name="defaultPivotStyle" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyle">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="tableStyleElement" type="CT_TableStyleElement" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="pivot" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="table" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableStyleElement">
        <xsd:attribute name="type" type="ST_TableStyleType" use="required"/>
        <xsd:attribute name="size" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="dxfId" type="ST_DxfId" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TableStyleType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="wholeTable"/>
            <xsd:enumeration value="headerRow"/>
            <xsd:enumeration value="totalRow"/>
            <xsd:enumeration value="firstColumn"/>
            <xsd:enumeration value="lastColumn"/>
            <xsd:enumeration value="firstRowStripe"/>
            <xsd:enumeration value="secondRowStripe"/>
            <xsd:enumeration value="firstColumnStripe"/>
            <xsd:enumeration value="secondColumnStripe"/>
            <xsd:enumeration value="firstHeaderCell"/>
            <xsd:enumeration value="lastHeaderCell"/>
            <xsd:enumeration value="firstTotalCell"/>
            <xsd:enumeration value="lastTotalCell"/>
            <xsd:enumeration value="firstSubtotalColumn"/>
            <xsd:enumeration value="secondSubtotalColumn"/>
            <xsd:enumeration value="thirdSubtotalColumn"/>
            <xsd:enumeration value="firstSubtotalRow"/>
            <xsd:enumeration value="secondSubtotalRow"/>
            <xsd:enumeration value="thirdSubtotalRow"/>
            <xsd:enumeration value="blankRow"/>
            <xsd:enumeration value="firstColumnSubheading"/>
            <xsd:enumeration value="secondColumnSubheading"/>
            <xsd:enumeration value="thirdColumnSubheading"/>
            <xsd:enumeration value="firstRowSubheading"/>
            <xsd:enumeration value="secondRowSubheading"/>
            <xsd:enumeration value="thirdRowSubheading"/>
            <xsd:enumeration value="pageFieldLabels"/>
            <xsd:enumeration value="pageFieldValues"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_BooleanProperty">
        <xsd:attribute name="val" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontSize">
        <xsd:attribute name="val" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_IntProperty">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="val" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontName">
        <xsd:attribute name="val" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VerticalAlignFontProperty">
        <xsd:attribute name="val" type="s:ST_VerticalAlignRun" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FontScheme">
        <xsd:attribute name="val" type="ST_FontScheme" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontScheme">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="major"/>
            <xsd:enumeration value="minor"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_UnderlineProperty">
        <xsd:attribute name="val" type="ST_UnderlineValues" use="optional" default="single"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_UnderlineValues">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="single"/>
            <xsd:enumeration value="double"/>
            <xsd:enumeration value="singleAccounting"/>
            <xsd:enumeration value="doubleAccounting"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FontFamily">
        <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FontFamily">
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="14"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Font">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="name" type="CT_FontName" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="charset" type="CT_IntProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="family" type="CT_FontFamily" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="b" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="i" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="strike" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outline" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="shadow" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="condense" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extend" type="CT_BooleanProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="color" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sz" type="CT_FontSize" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="u" type="CT_UnderlineProperty" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="vertAlign" type="CT_VerticalAlignFontProperty" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="scheme" type="CT_FontScheme" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:attributeGroup name="AG_AutoFormat">
        <xsd:attribute name="autoFormatId" type="xsd:unsignedInt"/>
        <xsd:attribute name="applyNumberFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyBorderFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyFontFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyPatternFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyAlignmentFormats" type="xsd:boolean"/>
        <xsd:attribute name="applyWidthHeightFormats" type="xsd:boolean"/>
    </xsd:attributeGroup>
    <xsd:element name="externalLink" type="CT_ExternalLink"/>
    <xsd:complexType name="CT_ExternalLink">
      <xsd:sequence>
        <xsd:choice>
            <xsd:element name="externalBook" type="CT_ExternalBook" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ddeLink" type="CT_DdeLink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleLink" type="CT_OleLink" minOccurs="0" maxOccurs="1"/>
        </xsd:choice>
        <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalBook">
        <xsd:sequence>
            <xsd:element name="sheetNames" type="CT_ExternalSheetNames" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="definedNames" type="CT_ExternalDefinedNames" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="sheetDataSet" type="CT_ExternalSheetDataSet" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetNames">
        <xsd:sequence>
            <xsd:element name="sheetName" minOccurs="1" maxOccurs="unbounded"
              type="CT_ExternalSheetName"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetName">
        <xsd:attribute name="val" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalDefinedNames">
        <xsd:sequence>
            <xsd:element name="definedName" type="CT_ExternalDefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalDefinedName">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="refersTo" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetDataSet">
        <xsd:sequence>
            <xsd:element name="sheetData" type="CT_ExternalSheetData" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalSheetData">
        <xsd:sequence>
            <xsd:element name="row" type="CT_ExternalRow" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="refreshError" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalRow">
        <xsd:sequence>
            <xsd:element name="cell" type="CT_ExternalCell" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalCell">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_CellRef" use="optional"/>
        <xsd:attribute name="t" type="ST_CellType" use="optional" default="n"/>
        <xsd:attribute name="vm" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeLink">
        <xsd:sequence>
            <xsd:element name="ddeItems" type="CT_DdeItems" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="ddeService" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="ddeTopic" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeItems">
        <xsd:sequence>
            <xsd:element name="ddeItem" type="CT_DdeItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeItem">
        <xsd:sequence>
            <xsd:element name="values" type="CT_DdeValues" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" default="0"/>
        <xsd:attribute name="ole" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advise" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preferPic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeValues">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="value" minOccurs="1" maxOccurs="unbounded" type="CT_DdeValue"/>
        </xsd:sequence>
        <xsd:attribute name="rows" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="cols" type="xsd:unsignedInt" use="optional" default="1"/>
    </xsd:complexType>
    <xsd:complexType name="CT_DdeValue">
        <xsd:sequence>
            <xsd:element name="val" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_DdeValueType" use="optional" default="n"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_DdeValueType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="nil"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="str"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OleLink">
        <xsd:sequence>
            <xsd:element name="oleItems" type="CT_OleItems" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="progId" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleItems">
        <xsd:sequence>
            <xsd:element name="oleItem" type="CT_OleItem" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OleItem">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="icon" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advise" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="preferPic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="table" type="CT_Table"/>
    <xsd:complexType name="CT_Table">
        <xsd:sequence>
            <xsd:element name="autoFilter" type="CT_AutoFilter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sortState" type="CT_SortState" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tableColumns" type="CT_TableColumns" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tableStyleInfo" type="CT_TableStyleInfo" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="displayName" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
        <xsd:attribute name="tableType" type="ST_TableType" use="optional" default="worksheet"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="headerRowCount" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="insertRow" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="insertRowShift" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="totalsRowCount" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="totalsRowShown" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="published" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="headerRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="dataDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="tableBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowBorderDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="totalsRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="connectionId" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TableType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="worksheet"/>
            <xsd:enumeration value="xml"/>
            <xsd:enumeration value="queryTable"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TableStyleInfo">
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="showFirstColumn" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showLastColumn" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showRowStripes" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="showColumnStripes" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableColumns">
        <xsd:sequence>
            <xsd:element name="tableColumn" type="CT_TableColumn" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableColumn">
        <xsd:sequence>
            <xsd:element name="calculatedColumnFormula" type="CT_TableFormula" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="totalsRowFormula" type="CT_TableFormula" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="xmlColumnPr" type="CT_XmlColumnPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uniqueName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="totalsRowFunction" type="ST_TotalsRowFunction" use="optional"
          default="none"/>
        <xsd:attribute name="totalsRowLabel" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="queryTableFieldId" type="xsd:unsignedInt" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="headerRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="dataDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="totalsRowDxfId" type="ST_DxfId" use="optional"/>
        <xsd:attribute name="headerRowCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="dataCellStyle" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="totalsRowCellStyle" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TableFormula">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="array" type="xsd:boolean" default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:simpleType name="ST_TotalsRowFunction">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="min"/>
            <xsd:enumeration value="max"/>
            <xsd:enumeration value="average"/>
            <xsd:enumeration value="count"/>
            <xsd:enumeration value="countNums"/>
            <xsd:enumeration value="stdDev"/>
            <xsd:enumeration value="var"/>
            <xsd:enumeration value="custom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_XmlColumnPr">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mapId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="xpath" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="denormalized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="xmlDataType" type="ST_XmlDataType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_XmlDataType">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:element name="volTypes" type="CT_VolTypes"/>
    <xsd:complexType name="CT_VolTypes">
        <xsd:sequence>
            <xsd:element name="volType" type="CT_VolType" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_VolType">
        <xsd:sequence>
            <xsd:element name="main" type="CT_VolMain" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_VolDepType" use="required"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_VolMain">
        <xsd:sequence>
            <xsd:element name="tp" type="CT_VolTopic" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="first" type="s:ST_Xstring" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VolTopic">
        <xsd:sequence>
            <xsd:element name="v" type="s:ST_Xstring" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="stp" type="s:ST_Xstring" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tr" type="CT_VolTopicRef" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="t" type="ST_VolValueType" use="optional" default="n"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VolTopicRef">
        <xsd:attribute name="r" type="ST_CellRef" use="required"/>
        <xsd:attribute name="s" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_VolDepType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="realTimeData"/>
            <xsd:enumeration value="olapFunctions"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_VolValueType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="n"/>
            <xsd:enumeration value="e"/>
            <xsd:enumeration value="s"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="workbook" type="CT_Workbook"/>
    <xsd:complexType name="CT_Workbook">
        <xsd:sequence>
            <xsd:element name="fileVersion" type="CT_FileVersion" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fileSharing" type="CT_FileSharing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="workbookPr" type="CT_WorkbookPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="workbookProtection" type="CT_WorkbookProtection" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="bookViews" type="CT_BookViews" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sheets" type="CT_Sheets" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="functionGroups" type="CT_FunctionGroups" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="externalReferences" type="CT_ExternalReferences" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="definedNames" type="CT_DefinedNames" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="calcPr" type="CT_CalcPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="oleSize" type="CT_OleSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="customWorkbookViews" type="CT_CustomWorkbookViews" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="pivotCaches" type="CT_PivotCaches" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTagPr" type="CT_SmartTagPr" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="smartTagTypes" type="CT_SmartTagTypes" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="webPublishing" type="CT_WebPublishing" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fileRecoveryPr" type="CT_FileRecoveryPr" minOccurs="0"
              maxOccurs="unbounded"/>
            <xsd:element name="webPublishObjects" type="CT_WebPublishObjects" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileVersion">
        <xsd:attribute name="appName" type="xsd:string" use="optional"/>
        <xsd:attribute name="lastEdited" type="xsd:string" use="optional"/>
        <xsd:attribute name="lowestEdited" type="xsd:string" use="optional"/>
        <xsd:attribute name="rupBuild" type="xsd:string" use="optional"/>
        <xsd:attribute name="codeName" type="s:ST_Guid" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BookViews">
        <xsd:sequence>
            <xsd:element name="workbookView" type="CT_BookView" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BookView">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="visibility" type="ST_Visibility" use="optional" default="visible"/>
        <xsd:attribute name="minimized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showHorizontalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showVerticalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showSheetTabs" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="xWindow" type="xsd:int" use="optional"/>
        <xsd:attribute name="yWindow" type="xsd:int" use="optional"/>
        <xsd:attribute name="windowWidth" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="windowHeight" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="tabRatio" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="firstSheet" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="activeTab" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="autoFilterDateGrouping" type="xsd:boolean" use="optional"
          default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Visibility">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="visible"/>
            <xsd:enumeration value="hidden"/>
            <xsd:enumeration value="veryHidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_CustomWorkbookViews">
        <xsd:sequence>
            <xsd:element name="customWorkbookView" minOccurs="1" maxOccurs="unbounded"
              type="CT_CustomWorkbookView"/>
        </xsd:sequence>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:complexType name="CT_CustomWorkbookView">
        <xsd:sequence>
            <xsd:element name="extLst" minOccurs="0" type="CT_ExtensionList"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="guid" type="s:ST_Guid" use="required"/>
        <xsd:attribute name="autoUpdate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="mergeInterval" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="changesSavedWin" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="onlySync" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="personalView" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="includePrintSettings" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="includeHiddenRowCol" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="maximized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="minimized" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showHorizontalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showVerticalScroll" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showSheetTabs" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="xWindow" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="yWindow" type="xsd:int" use="optional" default="0"/>
        <xsd:attribute name="windowWidth" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="windowHeight" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="tabRatio" type="xsd:unsignedInt" use="optional" default="600"/>
        <xsd:attribute name="activeSheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="showFormulaBar" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showStatusbar" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showComments" type="ST_Comments" use="optional" default="commIndicator"/>
        <xsd:attribute name="showObjects" type="ST_Objects" use="optional" default="all"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Comments">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="commNone"/>
            <xsd:enumeration value="commIndicator"/>
            <xsd:enumeration value="commIndAndComment"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Objects">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="placeholders"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Sheets">
        <xsd:sequence>
            <xsd:element name="sheet" type="CT_Sheet" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Sheet">
        <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sheetId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="state" type="ST_SheetState" use="optional" default="visible"/>
        <xsd:attribute ref="r:id" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_SheetState">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="visible"/>
            <xsd:enumeration value="hidden"/>
            <xsd:enumeration value="veryHidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_WorkbookPr">
        <xsd:attribute name="date1904" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showObjects" type="ST_Objects" use="optional" default="all"/>
        <xsd:attribute name="showBorderUnselectedTables" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="filterPrivacy" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="promptedSolutions" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showInkAnnotation" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="backupFile" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveExternalLinkValues" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="updateLinks" type="ST_UpdateLinks" use="optional" default="userSet"/>
        <xsd:attribute name="codeName" type="xsd:string" use="optional"/>
        <xsd:attribute name="hidePivotFieldList" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showPivotChartFilter" type="xsd:boolean" default="false"/>
        <xsd:attribute name="allowRefreshQuery" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="publishItems" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="checkCompatibility" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoCompressPictures" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="refreshAllConnections" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="defaultThemeVersion" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_UpdateLinks">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="userSet"/>
            <xsd:enumeration value="never"/>
            <xsd:enumeration value="always"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SmartTagPr">
        <xsd:attribute name="embed" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="show" type="ST_SmartTagShow" use="optional" default="all"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SmartTagShow">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="all"/>
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="noIndicator"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SmartTagTypes">
        <xsd:sequence>
            <xsd:element name="smartTagType" type="CT_SmartTagType" minOccurs="0"
              maxOccurs="unbounded"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTagType">
        <xsd:attribute name="namespaceUri" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="name" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="url" type="s:ST_Xstring" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileRecoveryPr">
        <xsd:attribute name="autoRecover" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="crashSave" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="dataExtractLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="repairLoad" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CalcPr">
        <xsd:attribute name="calcId" type="xsd:unsignedInt"/>
        <xsd:attribute name="calcMode" type="ST_CalcMode" use="optional" default="auto"/>
        <xsd:attribute name="fullCalcOnLoad" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="refMode" type="ST_RefMode" use="optional" default="A1"/>
        <xsd:attribute name="iterate" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="iterateCount" type="xsd:unsignedInt" use="optional" default="100"/>
        <xsd:attribute name="iterateDelta" type="xsd:double" use="optional" default="0.001"/>
        <xsd:attribute name="fullPrecision" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="calcCompleted" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="calcOnSave" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="concurrentCalc" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="concurrentManualCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="forceFullCalc" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_CalcMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="manual"/>
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="autoNoTable"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_RefMode">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="A1"/>
            <xsd:enumeration value="R1C1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_DefinedNames">
        <xsd:sequence>
            <xsd:element name="definedName" type="CT_DefinedName" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DefinedName">
        <xsd:simpleContent>
            <xsd:extension base="ST_Formula">
                <xsd:attribute name="name" type="s:ST_Xstring" use="required"/>
                <xsd:attribute name="comment" type="s:ST_Xstring" use="optional"/>
                <xsd:attribute name="customMenu" type="s:ST_Xstring" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
              <xsd:attribute name="description" type="s:ST_Xstring" use="optional"/>
              <xsd:attribute name="help" type="s:ST_Xstring" use="optional"/>
              <xsd:attribute name="statusBar" type="s:ST_Xstring" use="optional"/>
              <xsd:attribute name="localSheetId" type="xsd:unsignedInt" use="optional"/>
              <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
              <xsd:attribute name="function" type="xsd:boolean" use="optional" default="false"/>
              <xsd:attribute name="vbProcedure" type="xsd:boolean" use="optional" default="false"/>
              <xsd:attribute name="xlm" type="xsd:boolean" use="optional" default="false"/>
              <xsd:attribute name="functionGroupId" type="xsd:unsignedInt" use="optional"/>
              <xsd:attribute name="shortcutKey" type="s:ST_Xstring" use="optional"/>
              <xsd:attribute name="publishToServer" type="xsd:boolean" use="optional"
                default="false"/>
              <xsd:attribute name="workbookParameter" type="xsd:boolean" use="optional"
                default="false"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalReferences">
        <xsd:sequence>
            <xsd:element name="externalReference" type="CT_ExternalReference" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExternalReference">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SheetBackgroundPicture">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCaches">
        <xsd:sequence>
            <xsd:element name="pivotCache" type="CT_PivotCache" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PivotCache">
        <xsd:attribute name="cacheId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FileSharing">
        <xsd:attribute name="readOnlyRecommended" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userName" type="s:ST_Xstring"/>
        <xsd:attribute name="algorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleSize">
        <xsd:attribute name="ref" type="ST_Ref" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WorkbookProtection">
        <xsd:attribute name="lockStructure" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lockWindows" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="lockRevision" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        <xsd:attribute name="revisionsAlgorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="revisionsHashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="revisionsSaltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="revisionsSpinCount" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="workbookAlgorithmName" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="workbookHashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="workbookSaltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="workbookSpinCount" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishing">
        <xsd:attribute name="css" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="thicket" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="longFileNames" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="vml" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="allowPng" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="targetScreenSize" type="ST_TargetScreenSize" use="optional"
          default="800x600"/>
        <xsd:attribute name="dpi" type="xsd:unsignedInt" use="optional" default="96"/>
        <xsd:attribute name="characterSet" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TargetScreenSize">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="544x376"/>
            <xsd:enumeration value="640x480"/>
            <xsd:enumeration value="720x512"/>
            <xsd:enumeration value="800x600"/>
            <xsd:enumeration value="1024x768"/>
            <xsd:enumeration value="1152x882"/>
            <xsd:enumeration value="1152x900"/>
            <xsd:enumeration value="1280x1024"/>
            <xsd:enumeration value="1600x1200"/>
            <xsd:enumeration value="1800x1440"/>
            <xsd:enumeration value="1920x1200"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FunctionGroups">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element name="functionGroup" type="CT_FunctionGroup" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="builtInGroupCount" type="xsd:unsignedInt" default="16" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FunctionGroup">
        <xsd:attribute name="name" type="s:ST_Xstring"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishObjects">
        <xsd:sequence>
            <xsd:element name="webPublishObject" type="CT_WebPublishObject" minOccurs="1"
              maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="count" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WebPublishObject">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="divId" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="sourceObject" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="destinationFile" type="s:ST_Xstring" use="required"/>
        <xsd:attribute name="title" type="s:ST_Xstring" use="optional"/>
        <xsd:attribute name="autoRepublish" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
</xsd:schema>
```

A.3 PresentationML

This schema is available in the file pml.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://purl.oclc.org/ooxml/presentationml/main"
  xmlns:p="http://purl.oclc.org/ooxml/presentationml/main"
  xmlns:a="http://purl.oclc.org/ooxml/drawingml/main"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes" elementFormDefault="qualified"
  targetNamespace="http://purl.oclc.org/ooxml/presentationml/main">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/main" schemaLocation="dml-main.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:simpleType name="ST_TransitionSideDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="u"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="d"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TransitionCornerDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="lu"/>
            <xsd:enumeration value="ru"/>
            <xsd:enumeration value="ld"/>
            <xsd:enumeration value="rd"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TransitionInOutDirectionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="out"/>
            <xsd:enumeration value="in"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SideDirectionTransition">
        <xsd:attribute name="dir" type="ST_TransitionSideDirectionType" use="optional" default="l"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CornerDirectionTransition">
        <xsd:attribute name="dir" type="ST_TransitionCornerDirectionType" use="optional"
          default="lu"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    <xsd:simpleType name="ST_TransitionEightDirectionType">
        <xsd:union memberTypes="ST_TransitionSideDirectionType ST_TransitionCornerDirectionType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_EightDirectionTransition">
        <xsd:attribute name="dir" type="ST_TransitionEightDirectionType" use="optional" default="l"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OrientationTransition">
        <xsd:attribute name="dir" type="ST_Direction" use="optional" default="horz"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InOutTransition">
        <xsd:attribute name="dir" type="ST_TransitionInOutDirectionType" use="optional"
          default="out"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OptionalBlackTransition">
        <xsd:attribute name="thruBlk" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SplitTransition">
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="horz"/>
        <xsd:attribute name="dir" type="ST_TransitionInOutDirectionType" use="optional"
          default="out"/>
    </xsd:complexType>
    <xsd:complexType name="CT_WheelTransition">
        <xsd:attribute name="spokes" type="xsd:unsignedInt" use="optional" default="4"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransitionStartSoundAction">
        <xsd:sequence>
            <xsd:element minOccurs="1" maxOccurs="1" name="snd" type="a:CT_EmbeddedWAVAudioFile"/>
        </xsd:sequence>
        <xsd:attribute name="loop" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransitionSoundAction">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="stSnd" type="CT_TransitionStartSoundAction"/>
            <xsd:element name="endSnd" type="CT_Empty"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TransitionSpeed">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="slow"/>
            <xsd:enumeration value="med"/>
            <xsd:enumeration value="fast"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideTransition">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="1">
                <xsd:element name="blinds" type="CT_OrientationTransition"/>
                <xsd:element name="checker" type="CT_OrientationTransition"/>
                <xsd:element name="circle" type="CT_Empty"/>
                <xsd:element name="dissolve" type="CT_Empty"/>
                <xsd:element name="comb" type="CT_OrientationTransition"/>
                <xsd:element name="cover" type="CT_EightDirectionTransition"/>
                <xsd:element name="cut" type="CT_OptionalBlackTransition"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
95              <xsd:element name="diamond" type="CT_Empty"/>
96              <xsd:element name="fade" type="CT_OptionalBlackTransition"/>
97              <xsd:element name="newsflash" type="CT_Empty"/>
98              <xsd:element name="plus" type="CT_Empty"/>
99              <xsd:element name="pull" type="CT_EightDirectionTransition"/>
100             <xsd:element name="push" type="CT_SideDirectionTransition"/>
101             <xsd:element name="random" type="CT_Empty"/>
102             <xsd:element name="randomBar" type="CT_OrientationTransition"/>
103             <xsd:element name="split" type="CT_SplitTransition"/>
104             <xsd:element name="strips" type="CT_CornerDirectionTransition"/>
105             <xsd:element name="wedge" type="CT_Empty"/>
106             <xsd:element name="wheel" type="CT_WheelTransition"/>
107             <xsd:element name="wipe" type="CT_SideDirectionTransition"/>
108             <xsd:element name="zoom" type="CT_InOutTransition"/>
109         </xsd:choice>
110         <xsd:element name="sndAc" minOccurs="0" maxOccurs="1" type="CT_TransitionSoundAction"/>
111         <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
112     </xsd:sequence>
113     <xsd:attribute name="spd" type="ST_TransitionSpeed" use="optional" default="fast"/>
114     <xsd:attribute name="advClick" type="xsd:boolean" use="optional" default="true"/>
115     <xsd:attribute name="advTm" type="xsd:unsignedInt" use="optional"/>
116 </xsd:complexType>
117 <xsd:simpleType name="ST_TLTimeIndefinite">
118     <xsd:restriction base="xsd:token">
119         <xsd:enumeration value="indefinite"/>
120     </xsd:restriction>
121 </xsd:simpleType>
122 <xsd:simpleType name="ST_TLTime">
123     <xsd:union memberTypes="xsd:unsignedInt ST_TLTimeIndefinite"/>
124 </xsd:simpleType>
125 <xsd:simpleType name="ST_TLTimeNodeID">
126     <xsd:restriction base="xsd:unsignedInt"/>
127 </xsd:simpleType>
128 <xsd:complexType name="CT_TLIterateIntervalTime">
129     <xsd:attribute name="val" type="ST_TLTime" use="required"/>
130 </xsd:complexType>
131 <xsd:complexType name="CT_TLIterateIntervalPercentage">
132     <xsd:attribute name="val" type="a:ST_PositivePercentage" use="required"/>
133 </xsd:complexType>
134 <xsd:simpleType name="ST_IterateType">
135     <xsd:restriction base="xsd:token">
136         <xsd:enumeration value="el"/>
137         <xsd:enumeration value="wd"/>
138         <xsd:enumeration value="lt"/>
139     </xsd:restriction>
140 </xsd:simpleType>
141 <xsd:complexType name="CT_TLIterateData">
142     <xsd:choice minOccurs="1" maxOccurs="1">
143         <xsd:element name="tmAbs" type="CT_TLIterateIntervalTime"/>
144         <xsd:element name="tmPct" type="CT_TLIterateIntervalPercentage"/>
145     </xsd:choice>
146     <xsd:attribute name="type" type="ST_IterateType" use="optional" default="el"/>
147     <xsd:attribute name="backwards" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_TLSubShapeId">
        <xsd:attribute name="spid" type="a:ST_ShapeID" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTextTargetElement">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="charRg" type="CT_IndexRange"/>
            <xsd:element name="pRg" type="CT_IndexRange"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLChartSubelementType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="gridLegend"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="ptInSeries"/>
            <xsd:enumeration value="ptInCategory"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLOleChartTargetElement">
        <xsd:attribute name="type" type="ST_TLChartSubelementType" use="required"/>
        <xsd:attribute name="lvl" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLShapeTargetElement">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="bg" type="CT_Empty"/>
            <xsd:element name="subSp" type="CT_TLSubShapeId"/>
            <xsd:element name="oleChartEl" type="CT_TLOleChartTargetElement"/>
            <xsd:element name="txEl" type="CT_TLTextTargetElement"/>
            <xsd:element name="graphicEl" type="a:CT_AnimationElementChoice"/>
        </xsd:choice>
        <xsd:attribute name="spid" type="a:ST_DrawingElementId" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeTargetElement">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="sldTgt" type="CT_Empty"/>
            <xsd:element name="sndTgt" type="a:CT_EmbeddedWAVAudioFile"/>
            <xsd:element name="spTgt" type="CT_TLShapeTargetElement"/>
            <xsd:element name="inkTgt" type="CT_TLSubShapeId"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTriggerTimeNodeID">
        <xsd:attribute name="val" type="ST_TLTimeNodeID" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTriggerRuntimeNode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="first"/>
            <xsd:enumeration value="last"/>
            <xsd:enumeration value="all"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTriggerRuntimeNode">
        <xsd:attribute name="val" type="ST_TLTriggerRuntimeNode" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTriggerEvent">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="onBegin"/>
            <xsd:enumeration value="onEnd"/>
            <xsd:enumeration value="begin"/>
            <xsd:enumeration value="end"/>
            <xsd:enumeration value="onClick"/>
            <xsd:enumeration value="onDblClick"/>
            <xsd:enumeration value="onMouseOver"/>
            <xsd:enumeration value="onMouseOut"/>
            <xsd:enumeration value="onNext"/>
            <xsd:enumeration value="onPrev"/>
            <xsd:enumeration value="onStopAudio"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeCondition">
        <xsd:choice minOccurs="0" maxOccurs="1">
            <xsd:element name="tgtEl" type="CT_TLTimeTargetElement"/>
            <xsd:element name="tn" type="CT_TLTriggerTimeNodeID"/>
            <xsd:element name="rtn" type="CT_TLTriggerRuntimeNode"/>
        </xsd:choice>
        <xsd:attribute name="evt" use="optional" type="ST_TLTriggerEvent"/>
        <xsd:attribute name="delay" type="ST_TLTime" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeConditionList">
        <xsd:sequence>
            <xsd:element name="cond" type="CT_TLTimeCondition" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TimeNodeList">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="par" type="CT_TLTimeNodeParallel"/>
            <xsd:element name="seq" type="CT_TLTimeNodeSequence"/>
            <xsd:element name="excl" type="CT_TLTimeNodeExclusive"/>
            <xsd:element name="anim" type="CT_TLAnimateBehavior"/>
            <xsd:element name="animClr" type="CT_TLAnimateColorBehavior"/>
            <xsd:element name="animEffect" type="CT_TLAnimateEffectBehavior"/>
            <xsd:element name="animMotion" type="CT_TLAnimateMotionBehavior"/>
            <xsd:element name="animRot" type="CT_TLAnimateRotationBehavior"/>
            <xsd:element name="animScale" type="CT_TLAnimateScaleBehavior"/>
            <xsd:element name="cmd" type="CT_TLCommandBehavior"/>
            <xsd:element name="set" type="CT_TLSetBehavior"/>
            <xsd:element name="audio" type="CT_TLMediaNodeAudio"/>
            <xsd:element name="video" type="CT_TLMediaNodeVideo"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTimeNodePresetClassType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="entr"/>
            <xsd:enumeration value="exit"/>
            <xsd:enumeration value="emph"/>
            <xsd:enumeration value="path"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="verb"/>
            <xsd:enumeration value="mediacall"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeRestartType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="always"/>
            <xsd:enumeration value="whenNotActive"/>
            <xsd:enumeration value="never"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeFillType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="remove"/>
            <xsd:enumeration value="freeze"/>
            <xsd:enumeration value="hold"/>
            <xsd:enumeration value="transition"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeSyncType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="canSlip"/>
            <xsd:enumeration value="locked"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeMasterRelation">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sameClick"/>
            <xsd:enumeration value="lastClick"/>
            <xsd:enumeration value="nextClick"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLTimeNodeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="clickEffect"/>
            <xsd:enumeration value="withEffect"/>
            <xsd:enumeration value="afterEffect"/>
            <xsd:enumeration value="mainSeq"/>
            <xsd:enumeration value="interactiveSeq"/>
            <xsd:enumeration value="clickPar"/>
            <xsd:enumeration value="withGroup"/>
            <xsd:enumeration value="afterGroup"/>
            <xsd:enumeration value="tmRoot"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommonTimeNodeData">
        <xsd:sequence>
            <xsd:element name="stCondLst" type="CT_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endCondLst" type="CT_TLTimeConditionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endSync" type="CT_TLTimeCondition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="iterate" type="CT_TLIterateData" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="childTnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="subTnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_TLTimeNodeID" use="optional"/>
        <xsd:attribute name="presetID" type="xsd:int" use="optional"/>
        <xsd:attribute name="presetClass" type="ST_TLTimeNodePresetClassType" use="optional"/>
        <xsd:attribute name="presetSubtype" type="xsd:int" use="optional"/>
        <xsd:attribute name="dur" type="ST_TLTime" use="optional"/>
        <xsd:attribute name="repeatCount" type="ST_TLTime" use="optional" default="1000"/>
        <xsd:attribute name="repeatDur" type="ST_TLTime" use="optional"/>
        <xsd:attribute name="spd" type="a:ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="accel" type="a:ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="decel" type="a:ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="autoRev" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="restart" type="ST_TLTimeNodeRestartType" use="optional"/>
        <xsd:attribute name="fill" type="ST_TLTimeNodeFillType" use="optional"/>
        <xsd:attribute name="syncBehavior" type="ST_TLTimeNodeSyncType" use="optional"/>
        <xsd:attribute name="tmFilter" type="xsd:string" use="optional"/>
        <xsd:attribute name="evtFilter" type="xsd:string" use="optional"/>
        <xsd:attribute name="display" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="masterRel" type="ST_TLTimeNodeMasterRelation" use="optional"/>
        <xsd:attribute name="bldLvl" type="xsd:int" use="optional"/>
        <xsd:attribute name="grpId" type="xsd:unsignedInt" use="optional"/>
        <xsd:attribute name="afterEffect" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="nodeType" type="ST_TLTimeNodeType" use="optional"/>
        <xsd:attribute name="nodePh" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeNodeParallel">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLNextActionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="seek"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLPreviousActionType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="skipTimed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeNodeSequence">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prevCondLst" type="CT_TLTimeConditionList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="nextCondLst" type="CT_TLTimeConditionList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="concurrent" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="prevAc" type="ST_TLPreviousActionType" use="optional"/>
        <xsd:attribute name="nextAc" type="ST_TLNextActionType" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeNodeExclusive">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TLBehaviorAttributeNameList">
        <xsd:sequence>
            <xsd:element name="attrName" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLBehaviorAdditiveType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="base"/>
            <xsd:enumeration value="sum"/>
            <xsd:enumeration value="repl"/>
            <xsd:enumeration value="mult"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorAccumulateType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="always"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorTransformType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="pt"/>
            <xsd:enumeration value="img"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLBehaviorOverrideType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="normal"/>
            <xsd:enumeration value="childStyle"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommonBehaviorData">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tgtEl" type="CT_TLTimeTargetElement" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="attrNameLst" type="CT_TLBehaviorAttributeNameList" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="additive" type="ST_TLBehaviorAdditiveType" use="optional"/>
        <xsd:attribute name="accumulate" type="ST_TLBehaviorAccumulateType" use="optional"/>
        <xsd:attribute name="xfrmType" type="ST_TLBehaviorTransformType" use="optional"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <xsd:attribute name="to" type="xsd:string" use="optional"/>
        <xsd:attribute name="by" type="xsd:string" use="optional"/>
        <xsd:attribute name="rctx" type="xsd:string" use="optional"/>
        <xsd:attribute name="override" type="ST_TLBehaviorOverrideType" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantBooleanVal">
        <xsd:attribute name="val" type="xsd:boolean" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantIntegerVal">
        <xsd:attribute name="val" type="xsd:int" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantFloatVal">
        <xsd:attribute name="val" type="xsd:float" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariantStringVal">
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimVariant">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="boolVal" type="CT_TLAnimVariantBooleanVal"/>
            <xsd:element name="intVal" type="CT_TLAnimVariantIntegerVal"/>
            <xsd:element name="fltVal" type="CT_TLAnimVariantFloatVal"/>
            <xsd:element name="strVal" type="CT_TLAnimVariantStringVal"/>
            <xsd:element name="clrVal" type="a:CT_Color"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLTimeAnimateValueTime">
        <xsd:union memberTypes="a:ST_PositiveFixedPercentage ST_TLTimeIndefinite"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLTimeAnimateValue">
        <xsd:sequence>
            <xsd:element name="val" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="tm" type="ST_TLTimeAnimateValueTime" use="optional"
          default="indefinite"/>
        <xsd:attribute name="fmla" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTimeAnimateValueList">
        <xsd:sequence>
            <xsd:element name="tav" type="CT_TLTimeAnimateValue" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateBehaviorCalcMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="discrete"/>
            <xsd:enumeration value="lin"/>
            <xsd:enumeration value="fmla"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateBehaviorValueType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="str"/>
            <xsd:enumeration value="num"/>
            <xsd:enumeration value="clr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateBehavior">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tavLst" type="CT_TLTimeAnimateValueList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="by" type="xsd:string" use="optional"/>
        <xsd:attribute name="from" type="xsd:string" use="optional"/>
        <xsd:attribute name="to" type="xsd:string" use="optional"/>
        <xsd:attribute name="calcmode" type="ST_TLAnimateBehaviorCalcMode" use="optional"/>
        <xsd:attribute name="valueType" type="ST_TLAnimateBehaviorValueType" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByRgbColorTransform">
        <xsd:attribute name="r" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="g" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="b" type="a:ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByHslColorTransform">
        <xsd:attribute name="h" type="a:ST_Angle" use="required"/>
        <xsd:attribute name="s" type="a:ST_FixedPercentage" use="required"/>
        <xsd:attribute name="l" type="a:ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLByAnimateColorTransform">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="rgb" type="CT_TLByRgbColorTransform"/>
            <xsd:element name="hsl" type="CT_TLByHslColorTransform"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateColorSpace">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="rgb"/>
            <xsd:enumeration value="hsl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateColorDirection">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="cw"/>
            <xsd:enumeration value="ccw"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateColorBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLByAnimateColorTransform" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="from" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="clrSpc" type="ST_TLAnimateColorSpace" use="optional"/>
        <xsd:attribute name="dir" type="ST_TLAnimateColorDirection" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateEffectTransition">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="in"/>
            <xsd:enumeration value="out"/>
            <xsd:enumeration value="none"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLAnimateEffectBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="progress" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="transition" type="ST_TLAnimateEffectTransition" use="optional"/>
        <xsd:attribute name="filter" type="xsd:string" use="optional"/>
        <xsd:attribute name="prLst" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLAnimateMotionBehaviorOrigin">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="parent"/>
            <xsd:enumeration value="layout"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_TLAnimateMotionPathEditMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="relative"/>
            <xsd:enumeration value="fixed"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLPoint">
        <xsd:attribute name="x" type="a:ST_Percentage" use="required"/>
        <xsd:attribute name="y" type="a:ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateMotionBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="from" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="rCtr" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="origin" type="ST_TLAnimateMotionBehaviorOrigin" use="optional"/>
        <xsd:attribute name="path" type="xsd:string" use="optional"/>
        <xsd:attribute name="pathEditMode" type="ST_TLAnimateMotionPathEditMode" use="optional"/>
        <xsd:attribute name="rAng" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="ptsTypes" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateRotationBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="by" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="from" type="a:ST_Angle" use="optional"/>
        <xsd:attribute name="to" type="a:ST_Angle" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLAnimateScaleBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="by" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="from" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLPoint" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="zoomContents" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLCommandType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="evt"/>
            <xsd:enumeration value="call"/>
            <xsd:enumeration value="verb"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLCommandBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute type="ST_TLCommandType" name="type" use="optional"/>
        <xsd:attribute name="cmd" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLSetBehavior">
        <xsd:sequence>
            <xsd:element name="cBhvr" type="CT_TLCommonBehaviorData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="to" type="CT_TLAnimVariant" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_TLCommonMediaNodeData">
        <xsd:sequence>
            <xsd:element name="cTn" type="CT_TLCommonTimeNodeData" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tgtEl" type="CT_TLTimeTargetElement" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="vol" type="a:ST_PositiveFixedPercentage" default="50%" use="optional"/>
        <xsd:attribute name="mute" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="numSld" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="showWhenStopped" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLMediaNodeAudio">
        <xsd:sequence>
            <xsd:element name="cMediaNode" type="CT_TLCommonMediaNodeData" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isNarration" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLMediaNodeVideo">
        <xsd:sequence>
            <xsd:element name="cMediaNode" type="CT_TLCommonMediaNodeData" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="fullScrn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:attributeGroup name="AG_TLBuild">
        <xsd:attribute name="spid" type="a:ST_DrawingElementId" use="required"/>
        <xsd:attribute name="grpId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="uiExpand" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:attributeGroup>
    <xsd:complexType name="CT_TLTemplate">
        <xsd:sequence>
            <xsd:element name="tnLst" type="CT_TimeNodeList" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="lvl" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLTemplateList">
        <xsd:sequence>
            <xsd:element name="tmpl" type="CT_TLTemplate" minOccurs="0" maxOccurs="9"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLParaBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
            <xsd:enumeration value="p"/>
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="whole"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLBuildParagraph">
        <xsd:sequence>
            <xsd:element name="tmplLst" type="CT_TLTemplateList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="build" type="ST_TLParaBuildType" use="optional" default="whole"/>
        <xsd:attribute name="bldLvl" type="xsd:unsignedInt" use="optional" default="1"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoUpdateAnimBg" type="xsd:boolean" default="true" use="optional"/>
        <xsd:attribute name="rev" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="advAuto" type="ST_TLTime" use="optional" default="indefinite"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLDiagramBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="whole"/>
            <xsd:enumeration value="depthByNode"/>
            <xsd:enumeration value="depthByBranch"/>
            <xsd:enumeration value="breadthByNode"/>
            <xsd:enumeration value="breadthByLvl"/>
            <xsd:enumeration value="cw"/>
            <xsd:enumeration value="cwIn"/>
            <xsd:enumeration value="cwOut"/>
            <xsd:enumeration value="ccw"/>
            <xsd:enumeration value="ccwIn"/>
            <xsd:enumeration value="ccwOut"/>
            <xsd:enumeration value="inByRing"/>
            <xsd:enumeration value="outByRing"/>
            <xsd:enumeration value="up"/>
            <xsd:enumeration value="down"/>
            <xsd:enumeration value="allAtOnce"/>
            <xsd:enumeration value="cust"/>
        </xsd:restriction>
    </xsd:simpleType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_TLBuildDiagram">
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="bld" type="ST_TLDiagramBuildType" use="optional" default="whole"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_TLOleChartBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="seriesEl"/>
            <xsd:enumeration value="categoryEl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_TLOleBuildChart">
        <xsd:attributeGroup ref="AG_TLBuild"/>
        <xsd:attribute name="bld" type="ST_TLOleChartBuildType" use="optional" default="allAtOnce"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TLGraphicalObjectBuild">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="bldAsOne" type="CT_Empty"/>
            <xsd:element name="bldSub" type="a:CT_AnimationGraphicalObjectBuildProperties"/>
        </xsd:choice>
        <xsd:attributeGroup ref="AG_TLBuild"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BuildList">
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="bldP" type="CT_TLBuildParagraph"/>
            <xsd:element name="bldDgm" type="CT_TLBuildDiagram"/>
            <xsd:element name="bldOleChart" type="CT_TLOleBuildChart"/>
            <xsd:element name="bldGraphic" type="CT_TLGraphicalObjectBuild"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideTiming">
        <xsd:sequence>
            <xsd:element name="tnLst" type="CT_TimeNodeList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bldLst" type="CT_BuildList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Empty"/>
    <xsd:simpleType name="ST_Name">
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Direction">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Index">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_IndexRange">
        <xsd:attribute name="st" type="ST_Index" use="required"/>
        <xsd:attribute name="end" type="ST_Index" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideRelationshipListEntry">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideRelationshipList">
        <xsd:sequence>
            <xsd:element name="sld" type="CT_SlideRelationshipListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShowId">
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_SlideListChoice">
        <xsd:choice>
            <xsd:element name="sldAll" type="CT_Empty"/>
            <xsd:element name="sldRg" type="CT_IndexRange"/>
            <xsd:element name="custShow" type="CT_CustomShowId"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_CustomerData">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TagsData">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomerDataList">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="custData" type="CT_CustomerData" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="tags" type="CT_TagsData" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Extension">
        <xsd:sequence>
            <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_ExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_ExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ExtensionListModify">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:group ref="EG_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="mod" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentAuthor">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="name" type="ST_Name" use="required"/>
        <xsd:attribute name="initials" type="ST_Name" use="required"/>
        <xsd:attribute name="lastIdx" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="clrIdx" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentAuthorList">
        <xsd:sequence>
            <xsd:element name="cmAuthor" type="CT_CommentAuthor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="cmAuthorLst" type="CT_CommentAuthorList"/>
    <xsd:complexType name="CT_Comment">
        <xsd:sequence>
            <xsd:element name="pos" type="a:CT_Point2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="text" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="authorId" type="xsd:unsignedInt" use="required"/>
        <xsd:attribute name="dt" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="idx" type="ST_Index" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommentList">
        <xsd:sequence>
            <xsd:element name="cm" type="CT_Comment" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="cmLst" type="CT_CommentList"/>
    <xsd:attributeGroup name="AG_Ole">
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="showAsIcon" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="imgW" type="a:ST_PositiveCoordinate32" use="optional"/>
        <xsd:attribute name="imgH" type="a:ST_PositiveCoordinate32" use="optional"/>
    </xsd:attributeGroup>
    <xsd:simpleType name="ST_OleObjectFollowColorScheme">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="full"/>
            <xsd:enumeration value="textAndBackground"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_OleObjectEmbed">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
        <xsd:attribute name="followColorScheme" type="ST_OleObjectFollowColorScheme" use="optional"
          default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObjectLink">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="updateAutomatic" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OleObject">
        <xsd:sequence>
            <xsd:choice minOccurs="1" maxOccurs="1">
              <xsd:element name="embed" type="CT_OleObjectEmbed"/>
              <xsd:element name="link" type="CT_OleObjectLink"/>
            </xsd:choice>
            <xsd:element name="pic" type="CT_Picture" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Ole"/>
        <xsd:attribute name="progId" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:element name="oleObj" type="CT_OleObject"/>
    <xsd:complexType name="CT_Control">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="pic" type="CT_Picture" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Ole"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ControlList">
        <xsd:sequence>
            <xsd:element name="control" type="CT_Control" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideId">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="256"/>
            <xsd:maxExclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideId" use="required"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideIdList">
        <xsd:sequence>
            <xsd:element name="sldId" type="CT_SlideIdListEntry" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideMasterId">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideMasterId" use="optional"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideMasterIdList">
        <xsd:sequence>
            <xsd:element name="sldMasterId" type="CT_SlideMasterIdListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesMasterIdList">
        <xsd:sequence>
            <xsd:element name="notesMasterId" type="CT_NotesMasterIdListEntry" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_HandoutMasterIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HandoutMasterIdList">
        <xsd:sequence>
            <xsd:element name="handoutMasterId" type="CT_HandoutMasterIdListEntry" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EmbeddedFontDataId">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EmbeddedFontListEntry">
        <xsd:sequence>
            <xsd:element name="font" type="a:CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="regular" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bold" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="italic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="boldItalic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_EmbeddedFontList">
        <xsd:sequence>
            <xsd:element name="embeddedFont" type="CT_EmbeddedFontListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SmartTags">
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShow">
        <xsd:sequence>
            <xsd:element name="sldLst" type="CT_SlideRelationshipList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="ST_Name" use="required"/>
        <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomShowList">
        <xsd:sequence>
            <xsd:element name="custShow" type="CT_CustomShow" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_PhotoAlbumLayout">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="fitToSlide"/>
            <xsd:enumeration value="1pic"/>
            <xsd:enumeration value="2pic"/>
            <xsd:enumeration value="4pic"/>
            <xsd:enumeration value="1picTitle"/>
            <xsd:enumeration value="2picTitle"/>
            <xsd:enumeration value="4picTitle"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PhotoAlbumFrameShape">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="frameStyle1"/>
            <xsd:enumeration value="frameStyle2"/>
            <xsd:enumeration value="frameStyle3"/>
            <xsd:enumeration value="frameStyle4"/>
            <xsd:enumeration value="frameStyle5"/>
            <xsd:enumeration value="frameStyle6"/>
            <xsd:enumeration value="frameStyle7"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PhotoAlbum">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bw" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showCaptions" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="layout" type="ST_PhotoAlbumLayout" use="optional" default="fitToSlide"/>
        <xsd:attribute name="frame" type="ST_PhotoAlbumFrameShape" use="optional"
          default="frameStyle1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideSizeCoordinate">
        <xsd:restriction base="a:ST_PositiveCoordinate32">
            <xsd:minInclusive value="914400"/>
            <xsd:maxInclusive value="51206400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_SlideSizeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="screen4x3"/>
            <xsd:enumeration value="letter"/>
            <xsd:enumeration value="A4"/>
            <xsd:enumeration value="35mm"/>
            <xsd:enumeration value="overhead"/>
            <xsd:enumeration value="banner"/>
            <xsd:enumeration value="custom"/>
            <xsd:enumeration value="ledger"/>
            <xsd:enumeration value="A3"/>
            <xsd:enumeration value="B4ISO"/>
            <xsd:enumeration value="B5ISO"/>
            <xsd:enumeration value="B4JIS"/>
            <xsd:enumeration value="B5JIS"/>
            <xsd:enumeration value="hagakiCard"/>
            <xsd:enumeration value="screen16x9"/>
            <xsd:enumeration value="screen16x10"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideSize">
        <xsd:attribute name="cx" type="ST_SlideSizeCoordinate" use="required"/>
        <xsd:attribute name="cy" type="ST_SlideSizeCoordinate" use="required"/>
        <xsd:attribute name="type" type="ST_SlideSizeType" use="optional" default="custom"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Kinsoku">
        <xsd:attribute name="lang" type="xsd:string" use="optional"/>
        <xsd:attribute name="invalStChars" type="xsd:string" use="required"/>
        <xsd:attribute name="invalEndChars" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_BookmarkIdSeed">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
            <xsd:maxExclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ModifyVerifier">
        <xsd:attribute name="algorithmName" type="xsd:string" use="optional"/>
        <xsd:attribute name="hashValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="saltValue" type="xsd:base64Binary" use="optional"/>
        <xsd:attribute name="spinValue" type="xsd:unsignedInt" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Presentation">
        <xsd:sequence>
            <xsd:element name="sldMasterIdLst" type="CT_SlideMasterIdList" minOccurs="0"
              maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="notesMasterIdLst" type="CT_NotesMasterIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="handoutMasterIdLst" type="CT_HandoutMasterIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="sldIdLst" type="CT_SlideIdList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sldSz" type="CT_SlideSize" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notesSz" type="a:CT_PositiveSize2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="smartTags" type="CT_SmartTags" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="embeddedFontLst" type="CT_EmbeddedFontList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="custShowLst" type="CT_CustomShowList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="photoAlbum" type="CT_PhotoAlbum" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="kinsoku" type="CT_Kinsoku" minOccurs="0"/>
            <xsd:element name="defaultTextStyle" type="a:CT_TextListStyle" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="modifyVerifier" type="CT_ModifyVerifier" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="serverZoom" type="a:ST_Percentage" use="optional" default="50%"/>
        <xsd:attribute name="firstSlideNum" type="xsd:int" use="optional" default="1"/>
        <xsd:attribute name="showSpecialPlsOnTitleSld" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="rtl" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="removePersonalInfoOnSave" type="xsd:boolean" use="optional"
          default="false"/>
        <xsd:attribute name="compatMode" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="strictFirstAndLastChars" type="xsd:boolean" use="optional"
          default="true"/>
        <xsd:attribute name="embedTrueTypeFonts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="saveSubsetFonts" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="autoCompressPictures" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="bookmarkIdSeed" type="ST_BookmarkIdSeed" use="optional" default="1"/>
        <xsd:attribute name="conformance" type="s:ST_ConformanceClass"/>
    </xsd:complexType>
    <xsd:element name="presentation" type="CT_Presentation"/>
    <xsd:complexType name="CT_HtmlPublishProperties">
        <xsd:sequence>
            <xsd:group ref="EG_SlideListChoice" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showSpeakerNotes" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="target" type="xsd:string" use="optional"/>
        <xsd:attribute name="title" type="xsd:string" use="optional" default=""/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PrintWhat">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="slides"/>
            <xsd:enumeration value="handouts1"/>
            <xsd:enumeration value="handouts2"/>
            <xsd:enumeration value="handouts3"/>
            <xsd:enumeration value="handouts4"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="handouts6"/>
            <xsd:enumeration value="handouts9"/>
            <xsd:enumeration value="notes"/>
            <xsd:enumeration value="outline"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PrintColorMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bw"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="clr"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PrintProperties">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prnWhat" type="ST_PrintWhat" use="optional" default="slides"/>
        <xsd:attribute name="clrMode" type="ST_PrintColorMode" use="optional" default="clr"/>
        <xsd:attribute name="hiddenSlides" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="scaleToFitPaper" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="frameSlides" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShowInfoBrowse">
        <xsd:attribute name="showScrollbar" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShowInfoKiosk">
        <xsd:attribute name="restart" type="xsd:unsignedInt" use="optional" default="300000"/>
    </xsd:complexType>
    <xsd:group name="EG_ShowType">
        <xsd:choice>
            <xsd:element name="present" type="CT_Empty"/>
            <xsd:element name="browse" type="CT_ShowInfoBrowse"/>
            <xsd:element name="kiosk" type="CT_ShowInfoKiosk"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_ShowProperties">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:group ref="EG_ShowType" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_SlideListChoice" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="penClr" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="loop" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showNarration" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showAnimation" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="useTimings" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PresentationProperties">
        <xsd:sequence>
            <xsd:element name="prnPr" type="CT_PrintProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="showPr" type="CT_ShowProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="clrMru" type="a:CT_ColorMRU" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="presentationPr" type="CT_PresentationProperties"/>
    <xsd:complexType name="CT_HeaderFooter">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="sldNum" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="hdr" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="ftr" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="dt" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PlaceholderType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="title"/>
            <xsd:enumeration value="body"/>
            <xsd:enumeration value="ctrTitle"/>
            <xsd:enumeration value="subTitle"/>
            <xsd:enumeration value="dt"/>
            <xsd:enumeration value="sldNum"/>
            <xsd:enumeration value="ftr"/>
            <xsd:enumeration value="hdr"/>
            <xsd:enumeration value="obj"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="tbl"/>
            <xsd:enumeration value="clipArt"/>
            <xsd:enumeration value="dgm"/>
            <xsd:enumeration value="media"/>
            <xsd:enumeration value="sldImg"/>
            <xsd:enumeration value="pic"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PlaceholderSize">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="full"/>
            <xsd:enumeration value="half"/>
            <xsd:enumeration value="quarter"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Placeholder">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="ST_PlaceholderType" use="optional" default="obj"/>
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="horz"/>
        <xsd:attribute name="sz" type="ST_PlaceholderSize" use="optional" default="full"/>
        <xsd:attribute name="idx" type="xsd:unsignedInt" use="optional" default="0"/>
        <xsd:attribute name="hasCustomPrompt" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ApplicationNonVisualDrawingProps">
        <xsd:sequence>
            <xsd:element name="ph" type="CT_Placeholder" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:group ref="a:EG_Media" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isPhoto" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userDrawn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="a:CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Shape">
        <xsd:sequence>
            <xsd:element name="nvSpPr" type="CT_ShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txBody" type="a:CT_TextBody" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="useBgFill" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="a:CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Connector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_ConnectorNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="a:CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Picture">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_PictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="a:CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="a:CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="a:CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="a:CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GraphicalObjectFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="xfrm" type="a:CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="a:graphic" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="a:ST_BlackWhiteMode" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="a:CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="a:CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="nvPr" type="CT_ApplicationNonVisualDrawingProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GroupShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="a:CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="sp" type="CT_Shape"/>
                <xsd:element name="grpSp" type="CT_GroupShape"/>
                <xsd:element name="graphicFrame" type="CT_GraphicalObjectFrame"/>
                <xsd:element name="cxnSp" type="CT_Connector"/>
                <xsd:element name="pic" type="CT_Picture"/>
                <xsd:element name="contentPart" type="CT_Rel"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Rel">
        <xsd:attribute ref="r:id" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
    </xsd:complexType>
    <xsd:group name="EG_TopLevelSlide">
        <xsd:sequence>
            <xsd:element name="clrMap" type="a:CT_ColorMapping" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="EG_ChildSlide">
        <xsd:sequence>
            <xsd:element name="clrMapOvr" type="a:CT_ColorMappingOverride" minOccurs="0"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="AG_ChildSlide">
        <xsd:attribute name="showMasterSp" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="showMasterPhAnim" type="xsd:boolean" use="optional" default="true"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_BackgroundProperties">
        <xsd:sequence>
            <xsd:group ref="a:EG_FillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="a:EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="shadeToTitle" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:group name="EG_Background">
        <xsd:choice>
            <xsd:element name="bgPr" type="CT_BackgroundProperties"/>
            <xsd:element name="bgRef" type="a:CT_StyleMatrixReference"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Background">
        <xsd:sequence>
            <xsd:group ref="EG_Background"/>
        </xsd:sequence>
        <xsd:attribute name="bwMode" type="a:ST_BlackWhiteMode" use="optional" default="white"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonSlideData">
        <xsd:sequence>
            <xsd:element name="bg" type="CT_Background" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="spTree" type="CT_GroupShape" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="custDataLst" type="CT_CustomerDataList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="controls" type="CT_ControlList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_Slide">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
        <xsd:attribute name="show" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:element name="sld" type="CT_Slide"/>
    <xsd:simpleType name="ST_SlideLayoutType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="title"/>
            <xsd:enumeration value="tx"/>
            <xsd:enumeration value="twoColTx"/>
            <xsd:enumeration value="tbl"/>
            <xsd:enumeration value="txAndChart"/>
            <xsd:enumeration value="chartAndTx"/>
            <xsd:enumeration value="dgm"/>
            <xsd:enumeration value="chart"/>
            <xsd:enumeration value="txAndClipArt"/>
            <xsd:enumeration value="clipArtAndTx"/>
            <xsd:enumeration value="titleOnly"/>
            <xsd:enumeration value="blank"/>
            <xsd:enumeration value="txAndObj"/>
            <xsd:enumeration value="objAndTx"/>
            <xsd:enumeration value="objOnly"/>
            <xsd:enumeration value="obj"/>
            <xsd:enumeration value="txAndMedia"/>
            <xsd:enumeration value="mediaAndTx"/>
            <xsd:enumeration value="objOverTx"/>
            <xsd:enumeration value="txOverObj"/>
            <xsd:enumeration value="txAndTwoObj"/>
            <xsd:enumeration value="twoObjAndTx"/>
            <xsd:enumeration value="twoObjOverTx"/>
            <xsd:enumeration value="fourObj"/>
            <xsd:enumeration value="vertTx"/>
            <xsd:enumeration value="clipArtAndVertTx"/>
            <xsd:enumeration value="vertTitleAndTx"/>
            <xsd:enumeration value="vertTitleAndTxOverChart"/>
            <xsd:enumeration value="twoObj"/>
            <xsd:enumeration value="objAndTwoObj"/>
            <xsd:enumeration value="twoObjAndObj"/>
            <xsd:enumeration value="cust"/>
            <xsd:enumeration value="secHead"/>
            <xsd:enumeration value="twoTxTwoObj"/>
            <xsd:enumeration value="objTx"/>
            <xsd:enumeration value="picTx"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideLayout">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
        <xsd:attribute name="matchingName" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="type" type="ST_SlideLayoutType" use="optional" default="cust"/>
        <xsd:attribute name="preserve" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="userDrawn" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="sldLayout" type="CT_SlideLayout"/>
    <xsd:complexType name="CT_SlideMasterTextStyles">
        <xsd:sequence>
            <xsd:element name="titleStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bodyStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="otherStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_SlideLayoutId">
        <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="2147483648"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SlideLayoutIdListEntry">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_SlideLayoutId" use="optional"/>
        <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideLayoutIdList">
        <xsd:sequence>
            <xsd:element name="sldLayoutId" type="CT_SlideLayoutIdListEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideMaster">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sldLayoutIdLst" type="CT_SlideLayoutIdList" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="transition" type="CT_SlideTransition" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="timing" type="CT_SlideTiming" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="txStyles" type="CT_SlideMasterTextStyles" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="preserve" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:element name="sldMaster" type="CT_SlideMaster"/>
    <xsd:complexType name="CT_HandoutMaster">
        <xsd:sequence>
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="handoutMaster" type="CT_HandoutMaster"/>
    <xsd:complexType name="CT_NotesMaster">
        <xsd:sequence>
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_TopLevelSlide" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hf" type="CT_HeaderFooter" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="notesStyle" type="a:CT_TextListStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="notesMaster" type="CT_NotesMaster"/>
    <xsd:complexType name="CT_NotesSlide">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cSld" type="CT_CommonSlideData" minOccurs="1" maxOccurs="1"/>
            <xsd:group ref="EG_ChildSlide" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionListModify" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_ChildSlide"/>
    </xsd:complexType>
    <xsd:element name="notes" type="CT_NotesSlide"/>
    <xsd:complexType name="CT_SlideSyncProperties">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="serverSldId" type="xsd:string" use="required"/>
        <xsd:attribute name="serverSldModifiedTime" type="xsd:dateTime" use="required"/>
        <xsd:attribute name="clientInsertedTime" type="xsd:dateTime" use="required"/>
    </xsd:complexType>
    <xsd:element name="sldSyncPr" type="CT_SlideSyncProperties"/>
    <xsd:complexType name="CT_StringTag">
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="val" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TagList">
        <xsd:sequence>
            <xsd:element name="tag" type="CT_StringTag" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="tagLst" type="CT_TagList"/>
    <xsd:simpleType name="ST_SplitterBarState">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="minimized"/>
            <xsd:enumeration value="restored"/>
            <xsd:enumeration value="maximized"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ViewType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sldView"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="sldMasterView"/>
            <xsd:enumeration value="notesView"/>
            <xsd:enumeration value="handoutView"/>
            <xsd:enumeration value="notesMasterView"/>
            <xsd:enumeration value="outlineView"/>
            <xsd:enumeration value="sldSorterView"/>
            <xsd:enumeration value="sldThumbnailView"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_NormalViewPortion">
        <xsd:attribute name="sz" type="a:ST_PositiveFixedPercentage" use="required"/>
        <xsd:attribute name="autoAdjust" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NormalViewProperties">
        <xsd:sequence>
            <xsd:element name="restoredLeft" type="CT_NormalViewPortion" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="restoredTop" type="CT_NormalViewPortion" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showOutlineIcons" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="snapVertSplitter" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="vertBarState" type="ST_SplitterBarState" use="optional"
          default="restored"/>
        <xsd:attribute name="horzBarState" type="ST_SplitterBarState" use="optional"
          default="restored"/>
        <xsd:attribute name="preferSingleView" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonViewProperties">
        <xsd:sequence>
            <xsd:element name="scale" type="a:CT_Scale2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="origin" type="a:CT_Point2D" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="varScale" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesTextViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewSlideEntry">
        <xsd:attribute ref="r:id" use="required"/>
        <xsd:attribute name="collapse" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewSlideList">
        <xsd:sequence>
            <xsd:element name="sld" type="CT_OutlineViewSlideEntry" minOccurs="0"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OutlineViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="sldLst" type="CT_OutlineViewSlideList" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideSorterViewProperties">
        <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="showFormatting" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Guide">
        <xsd:attribute name="orient" type="ST_Direction" use="optional" default="vert"/>
        <xsd:attribute name="pos" type="a:ST_Coordinate32" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GuideList">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="guide" type="CT_Guide" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_CommonSlideViewProperties">
        <xsd:sequence>
            <xsd:element name="cViewPr" type="CT_CommonViewProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="guideLst" type="CT_GuideList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="snapToGrid" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="snapToObjects" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="showGuides" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SlideViewProperties">
        <xsd:sequence>
            <xsd:element name="cSldViewPr" type="CT_CommonSlideViewProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NotesViewProperties">
        <xsd:sequence>
            <xsd:element name="cSldViewPr" type="CT_CommonSlideViewProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ViewProperties">
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="normalViewPr" type="CT_NormalViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="slideViewPr" type="CT_SlideViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="outlineViewPr" type="CT_OutlineViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="notesTextViewPr" type="CT_NotesTextViewProperties" minOccurs="0"
              maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="sorterViewPr" type="CT_SlideSorterViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="notesViewPr" type="CT_NotesViewProperties" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="gridSpacing" type="a:CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="lastView" type="ST_ViewType" use="optional" default="sldView"/>
        <xsd:attribute name="showComments" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:element name="viewPr" type="CT_ViewProperties"/>
</xsd:schema>
```

A.4 DrawingML - Framework

A.4.1 DrawingML – Main

This schema is available in the file dml-main.xsd.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:r="http://purl.oclc.org/ooxml/officeDocument/relationships"
  xmlns:s="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
  xmlns="http://purl.oclc.org/ooxml/drawingml/main"
  targetNamespace="http://purl.oclc.org/ooxml/drawingml/main" elementFormDefault="qualified">
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/relationships"
      schemaLocation="shared-relationshipReference.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/officeDocument/sharedTypes"
      schemaLocation="shared-commonSimpleTypes.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/diagram" schemaLocation="dml-
      diagram.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/chart" schemaLocation="dml-
      chart.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/picture" schemaLocation="dml-
      picture.xsd"/>
    <xsd:import namespace="http://purl.oclc.org/ooxml/drawingml/lockedCanvas" schemaLocation="dml-
      lockedCanvas.xsd"/>
    <xsd:complexType name="CT_AudioFile">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
        <xsd:attribute name="contentType" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_VideoFile">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
        <xsd:attribute name="contentType" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_QuickTimeFile">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        </xsd:sequence>
        <xsd:attribute ref="r:link" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AudioCDTime">
        <xsd:attribute name="track" type="xsd:unsignedByte" use="required"/>
        <xsd:attribute name="time" type="xsd:unsignedInt" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AudioCD">
        <xsd:sequence>
            <xsd:element name="st" type="CT_AudioCDTime" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="end" type="CT_AudioCDTime" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_Media">
        <xsd:choice>
            <xsd:element name="audioCd" type="CT_AudioCD"/>
            <xsd:element name="wavAudioFile" type="CT_EmbeddedWAVAudioFile"/>
            <xsd:element name="audioFile" type="CT_AudioFile"/>
            <xsd:element name="videoFile" type="CT_VideoFile"/>
            <xsd:element name="quickTimeFile" type="CT_QuickTimeFile"/>
        </xsd:choice>
    </xsd:group>
    <xsd:element name="videoFile" type="CT_VideoFile"/>
    <xsd:simpleType name="ST_StyleMatrixColumnIndex">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FontCollectionIndex">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="major"/>
            <xsd:enumeration value="minor"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_ColorSchemeIndex">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="dk1"/>
            <xsd:enumeration value="lt1"/>
            <xsd:enumeration value="dk2"/>
            <xsd:enumeration value="lt2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hlink"/>
            <xsd:enumeration value="folHlink"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_ColorScheme">
        <xsd:sequence>
            <xsd:element name="dk1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="lt1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="dk2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lt2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent1" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent2" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent3" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent4" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent5" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="accent6" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hlink" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="folHlink" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_SupplementalFont">
        <xsd:attribute name="script" type="xsd:string" use="required"/>
        <xsd:attribute name="typeface" type="ST_TextTypeface" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_CustomColorList">
        <xsd:sequence>
            <xsd:element name="custClr" type="CT_CustomColor" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FontCollection">
        <xsd:sequence>
            <xsd:element name="latin" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="ea" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cs" type="CT_TextFont" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="font" type="CT_SupplementalFont" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectStyleItem">
        <xsd:sequence>
            <xsd:group ref="EG_EffectProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="scene3d" type="CT_Scene3D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="sp3d" type="CT_Shape3D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FontScheme">
        <xsd:sequence>
            <xsd:element name="majorFont" type="CT_FontCollection" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="minorFont" type="CT_FontCollection" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_FillStyleList">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LineStyleList">
        <xsd:sequence>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectStyleList">
        <xsd:sequence>
            <xsd:element name="effectStyle" type="CT_EffectStyleItem" minOccurs="3"
              maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BackgroundFillStyleList">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="3" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_StyleMatrix">
        <xsd:sequence>
            <xsd:element name="fillStyleLst" type="CT_FillStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lnStyleLst" type="CT_LineStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effectStyleLst" type="CT_EffectStyleList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="bgFillStyleLst" type="CT_BackgroundFillStyleList" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_BaseStyles">
        <xsd:sequence>
            <xsd:element name="clrScheme" type="CT_ColorScheme" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fontScheme" type="CT_FontScheme" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fmtScheme" type="CT_StyleMatrix" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_OfficeArtExtension">
        <xsd:sequence>
            <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Coordinate">
        <xsd:union memberTypes="ST_CoordinateUnqualified s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_CoordinateUnqualified">
        <xsd:restriction base="xsd:long">
            <xsd:minInclusive value="-27273042329600"/>
            <xsd:maxInclusive value="27273042316900"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Coordinate32">
        <xsd:union memberTypes="ST_Coordinate32Unqualified s:ST_UniversalMeasure"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Coordinate32Unqualified">
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveCoordinate">
        <xsd:restriction base="xsd:long">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="27273042316900"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveCoordinate32">
        <xsd:restriction base="ST_Coordinate32Unqualified">
            <xsd:minInclusive value="0"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_Angle">
        <xsd:restriction base="xsd:int"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Angle">
        <xsd:attribute name="val" type="ST_Angle" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FixedAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minExclusive value="-5400000"/>
            <xsd:maxExclusive value="5400000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_PositiveFixedAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minInclusive value="0"/>
            <xsd:maxExclusive value="21600000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositiveFixedAngle">
        <xsd:attribute name="val" type="ST_PositiveFixedAngle" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_Percentage">
        <xsd:union memberTypes="s:ST_Percentage"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_Percentage">
        <xsd:attribute name="val" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PositivePercentage">
        <xsd:union memberTypes="s:ST_PositivePercentage"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositivePercentage">
        <xsd:attribute name="val" type="ST_PositivePercentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_FixedPercentage">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:union memberTypes="s:ST_FixedPercentage"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_FixedPercentage">
        <xsd:attribute name="val" type="ST_FixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PositiveFixedPercentage">
        <xsd:union memberTypes="s:ST_PositiveFixedPercentage"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_PositiveFixedPercentage">
        <xsd:attribute name="val" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Ratio">
        <xsd:attribute name="n" type="xsd:long" use="required"/>
        <xsd:attribute name="d" type="xsd:long" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Point2D">
        <xsd:attribute name="x" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="y" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PositiveSize2D">
        <xsd:attribute name="cx" type="ST_PositiveCoordinate" use="required"/>
        <xsd:attribute name="cy" type="ST_PositiveCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ComplementTransform"/>
    <xsd:complexType name="CT_InverseTransform"/>
    <xsd:complexType name="CT_GrayscaleTransform"/>
    <xsd:complexType name="CT_GammaTransform"/>
    <xsd:complexType name="CT_InverseGammaTransform"/>
    <xsd:group name="EG_ColorTransform">
        <xsd:choice>
            <xsd:element name="tint" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="shade" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="comp" type="CT_ComplementTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="inv" type="CT_InverseTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gray" type="CT_GrayscaleTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alpha" type="CT_PositiveFixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaOff" type="CT_FixedPercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaMod" type="CT_PositivePercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hue" type="CT_PositiveFixedAngle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hueOff" type="CT_Angle" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hueMod" type="CT_PositivePercentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sat" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="satOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="satMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lum" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lumOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lumMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="red" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="redOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="redMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="green" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="greenOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="greenMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="blue" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blueOff" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blueMod" type="CT_Percentage" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="gamma" type="CT_GammaTransform" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="invGamma" type="CT_InverseGammaTransform" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_ScRgbColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="r" type="ST_Percentage" use="required"/>
        <xsd:attribute name="g" type="ST_Percentage" use="required"/>
        <xsd:attribute name="b" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SRgbColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="s:ST_HexColorRGB" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_HslColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="sat" type="ST_Percentage" use="required"/>
        <xsd:attribute name="lum" type="ST_Percentage" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SystemColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="scrollBar"/>
            <xsd:enumeration value="background"/>
            <xsd:enumeration value="activeCaption"/>
            <xsd:enumeration value="inactiveCaption"/>
            <xsd:enumeration value="menu"/>
            <xsd:enumeration value="window"/>
            <xsd:enumeration value="windowFrame"/>
            <xsd:enumeration value="menuText"/>
            <xsd:enumeration value="windowText"/>
            <xsd:enumeration value="captionText"/>
            <xsd:enumeration value="activeBorder"/>
            <xsd:enumeration value="inactiveBorder"/>
            <xsd:enumeration value="appWorkspace"/>
            <xsd:enumeration value="highlight"/>
            <xsd:enumeration value="highlightText"/>
            <xsd:enumeration value="btnFace"/>
            <xsd:enumeration value="btnShadow"/>
            <xsd:enumeration value="grayText"/>
            <xsd:enumeration value="btnText"/>
            <xsd:enumeration value="inactiveCaptionText"/>
            <xsd:enumeration value="btnHighlight"/>
            <xsd:enumeration value="3dDkShadow"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="3dLight"/>
            <xsd:enumeration value="infoText"/>
            <xsd:enumeration value="infoBk"/>
            <xsd:enumeration value="hotLight"/>
            <xsd:enumeration value="gradientActiveCaption"/>
            <xsd:enumeration value="gradientInactiveCaption"/>
            <xsd:enumeration value="menuHighlight"/>
            <xsd:enumeration value="menuBar"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SystemColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_SystemColorVal" use="required"/>
        <xsd:attribute name="lastClr" type="s:ST_HexColorRGB" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_SchemeColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="bg1"/>
            <xsd:enumeration value="tx1"/>
            <xsd:enumeration value="bg2"/>
            <xsd:enumeration value="tx2"/>
            <xsd:enumeration value="accent1"/>
            <xsd:enumeration value="accent2"/>
            <xsd:enumeration value="accent3"/>
            <xsd:enumeration value="accent4"/>
            <xsd:enumeration value="accent5"/>
            <xsd:enumeration value="accent6"/>
            <xsd:enumeration value="hlink"/>
            <xsd:enumeration value="folHlink"/>
            <xsd:enumeration value="phClr"/>
            <xsd:enumeration value="dk1"/>
            <xsd:enumeration value="lt1"/>
            <xsd:enumeration value="dk2"/>
            <xsd:enumeration value="lt2"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_SchemeColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_SchemeColorVal" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetColorVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="aliceBlue"/>
            <xsd:enumeration value="antiqueWhite"/>
            <xsd:enumeration value="aqua"/>
            <xsd:enumeration value="aquamarine"/>
            <xsd:enumeration value="azure"/>
            <xsd:enumeration value="beige"/>
            <xsd:enumeration value="bisque"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="blanchedAlmond"/>
            <xsd:enumeration value="blue"/>
            <xsd:enumeration value="blueViolet"/>
            <xsd:enumeration value="brown"/>
            <xsd:enumeration value="burlyWood"/>
            <xsd:enumeration value="cadetBlue"/>
            <xsd:enumeration value="chartreuse"/>
            <xsd:enumeration value="chocolate"/>
            <xsd:enumeration value="coral"/>
            <xsd:enumeration value="cornflowerBlue"/>
            <xsd:enumeration value="cornsilk"/>
            <xsd:enumeration value="crimson"/>
            <xsd:enumeration value="cyan"/>
            <xsd:enumeration value="darkBlue"/>
            <xsd:enumeration value="darkCyan"/>
            <xsd:enumeration value="darkGoldenrod"/>
            <xsd:enumeration value="darkGray"/>
            <xsd:enumeration value="darkGrey"/>
            <xsd:enumeration value="darkGreen"/>
            <xsd:enumeration value="darkKhaki"/>
            <xsd:enumeration value="darkMagenta"/>
            <xsd:enumeration value="darkOliveGreen"/>
            <xsd:enumeration value="darkOrange"/>
            <xsd:enumeration value="darkOrchid"/>
            <xsd:enumeration value="darkRed"/>
            <xsd:enumeration value="darkSalmon"/>
            <xsd:enumeration value="darkSeaGreen"/>
            <xsd:enumeration value="darkSlateBlue"/>
            <xsd:enumeration value="darkSlateGray"/>
            <xsd:enumeration value="darkSlateGrey"/>
            <xsd:enumeration value="darkTurquoise"/>
            <xsd:enumeration value="darkViolet"/>
            <xsd:enumeration value="dkBlue"/>
            <xsd:enumeration value="dkCyan"/>
            <xsd:enumeration value="dkGoldenrod"/>
            <xsd:enumeration value="dkGray"/>
            <xsd:enumeration value="dkGrey"/>
            <xsd:enumeration value="dkGreen"/>
            <xsd:enumeration value="dkKhaki"/>
            <xsd:enumeration value="dkMagenta"/>
            <xsd:enumeration value="dkOliveGreen"/>
            <xsd:enumeration value="dkOrange"/>
            <xsd:enumeration value="dkOrchid"/>
            <xsd:enumeration value="dkRed"/>
            <xsd:enumeration value="dkSalmon"/>
            <xsd:enumeration value="dkSeaGreen"/>
            <xsd:enumeration value="dkSlateBlue"/>
            <xsd:enumeration value="dkSlateGray"/>
            <xsd:enumeration value="dkSlateGrey"/>
            <xsd:enumeration value="dkTurquoise"/>
            <xsd:enumeration value="dkViolet"/>
            <xsd:enumeration value="deepPink"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="deepSkyBlue"/>
            <xsd:enumeration value="dimGray"/>
            <xsd:enumeration value="dimGrey"/>
            <xsd:enumeration value="dodgerBlue"/>
            <xsd:enumeration value="firebrick"/>
            <xsd:enumeration value="floralWhite"/>
            <xsd:enumeration value="forestGreen"/>
            <xsd:enumeration value="fuchsia"/>
            <xsd:enumeration value="gainsboro"/>
            <xsd:enumeration value="ghostWhite"/>
            <xsd:enumeration value="gold"/>
            <xsd:enumeration value="goldenrod"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="grey"/>
            <xsd:enumeration value="green"/>
            <xsd:enumeration value="greenYellow"/>
            <xsd:enumeration value="honeydew"/>
            <xsd:enumeration value="hotPink"/>
            <xsd:enumeration value="indianRed"/>
            <xsd:enumeration value="indigo"/>
            <xsd:enumeration value="ivory"/>
            <xsd:enumeration value="khaki"/>
            <xsd:enumeration value="lavender"/>
            <xsd:enumeration value="lavenderBlush"/>
            <xsd:enumeration value="lawnGreen"/>
            <xsd:enumeration value="lemonChiffon"/>
            <xsd:enumeration value="lightBlue"/>
            <xsd:enumeration value="lightCoral"/>
            <xsd:enumeration value="lightCyan"/>
            <xsd:enumeration value="lightGoldenrodYellow"/>
            <xsd:enumeration value="lightGray"/>
            <xsd:enumeration value="lightGrey"/>
            <xsd:enumeration value="lightGreen"/>
            <xsd:enumeration value="lightPink"/>
            <xsd:enumeration value="lightSalmon"/>
            <xsd:enumeration value="lightSeaGreen"/>
            <xsd:enumeration value="lightSkyBlue"/>
            <xsd:enumeration value="lightSlateGray"/>
            <xsd:enumeration value="lightSlateGrey"/>
            <xsd:enumeration value="lightSteelBlue"/>
            <xsd:enumeration value="lightYellow"/>
            <xsd:enumeration value="ltBlue"/>
            <xsd:enumeration value="ltCoral"/>
            <xsd:enumeration value="ltCyan"/>
            <xsd:enumeration value="ltGoldenrodYellow"/>
            <xsd:enumeration value="ltGray"/>
            <xsd:enumeration value="ltGrey"/>
            <xsd:enumeration value="ltGreen"/>
            <xsd:enumeration value="ltPink"/>
            <xsd:enumeration value="ltSalmon"/>
            <xsd:enumeration value="ltSeaGreen"/>
            <xsd:enumeration value="ltSkyBlue"/>
            <xsd:enumeration value="ltSlateGray"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="ltSlateGrey"/>
            <xsd:enumeration value="ltSteelBlue"/>
            <xsd:enumeration value="ltYellow"/>
            <xsd:enumeration value="lime"/>
            <xsd:enumeration value="limeGreen"/>
            <xsd:enumeration value="linen"/>
            <xsd:enumeration value="magenta"/>
            <xsd:enumeration value="maroon"/>
            <xsd:enumeration value="medAquamarine"/>
            <xsd:enumeration value="medBlue"/>
            <xsd:enumeration value="medOrchid"/>
            <xsd:enumeration value="medPurple"/>
            <xsd:enumeration value="medSeaGreen"/>
            <xsd:enumeration value="medSlateBlue"/>
            <xsd:enumeration value="medSpringGreen"/>
            <xsd:enumeration value="medTurquoise"/>
            <xsd:enumeration value="medVioletRed"/>
            <xsd:enumeration value="mediumAquamarine"/>
            <xsd:enumeration value="mediumBlue"/>
            <xsd:enumeration value="mediumOrchid"/>
            <xsd:enumeration value="mediumPurple"/>
            <xsd:enumeration value="mediumSeaGreen"/>
            <xsd:enumeration value="mediumSlateBlue"/>
            <xsd:enumeration value="mediumSpringGreen"/>
            <xsd:enumeration value="mediumTurquoise"/>
            <xsd:enumeration value="mediumVioletRed"/>
            <xsd:enumeration value="midnightBlue"/>
            <xsd:enumeration value="mintCream"/>
            <xsd:enumeration value="mistyRose"/>
            <xsd:enumeration value="moccasin"/>
            <xsd:enumeration value="navajoWhite"/>
            <xsd:enumeration value="navy"/>
            <xsd:enumeration value="oldLace"/>
            <xsd:enumeration value="olive"/>
            <xsd:enumeration value="oliveDrab"/>
            <xsd:enumeration value="orange"/>
            <xsd:enumeration value="orangeRed"/>
            <xsd:enumeration value="orchid"/>
            <xsd:enumeration value="paleGoldenrod"/>
            <xsd:enumeration value="paleGreen"/>
            <xsd:enumeration value="paleTurquoise"/>
            <xsd:enumeration value="paleVioletRed"/>
            <xsd:enumeration value="papayaWhip"/>
            <xsd:enumeration value="peachPuff"/>
            <xsd:enumeration value="peru"/>
            <xsd:enumeration value="pink"/>
            <xsd:enumeration value="plum"/>
            <xsd:enumeration value="powderBlue"/>
            <xsd:enumeration value="purple"/>
            <xsd:enumeration value="red"/>
            <xsd:enumeration value="rosyBrown"/>
            <xsd:enumeration value="royalBlue"/>
            <xsd:enumeration value="saddleBrown"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="salmon"/>
            <xsd:enumeration value="sandyBrown"/>
            <xsd:enumeration value="seaGreen"/>
            <xsd:enumeration value="seaShell"/>
            <xsd:enumeration value="sienna"/>
            <xsd:enumeration value="silver"/>
            <xsd:enumeration value="skyBlue"/>
            <xsd:enumeration value="slateBlue"/>
            <xsd:enumeration value="slateGray"/>
            <xsd:enumeration value="slateGrey"/>
            <xsd:enumeration value="snow"/>
            <xsd:enumeration value="springGreen"/>
            <xsd:enumeration value="steelBlue"/>
            <xsd:enumeration value="tan"/>
            <xsd:enumeration value="teal"/>
            <xsd:enumeration value="thistle"/>
            <xsd:enumeration value="tomato"/>
            <xsd:enumeration value="turquoise"/>
            <xsd:enumeration value="violet"/>
            <xsd:enumeration value="wheat"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="whiteSmoke"/>
            <xsd:enumeration value="yellow"/>
            <xsd:enumeration value="yellowGreen"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PresetColor">
        <xsd:sequence>
            <xsd:group ref="EG_ColorTransform" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="val" type="ST_PresetColorVal" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_OfficeArtExtensionList">
        <xsd:sequence>
            <xsd:element name="ext" type="CT_OfficeArtExtension" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="CT_OfficeArtExtensionList">
        <xsd:sequence>
            <xsd:group ref="EG_OfficeArtExtensionList" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Scale2D">
        <xsd:sequence>
            <xsd:element name="sx" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sy" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Transform2D">
        <xsd:sequence>
            <xsd:element name="off" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ext" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="rot" type="ST_Angle" use="optional" default="0"/>
        <xsd:attribute name="flipH" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="flipV" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupTransform2D">
        <xsd:sequence>
            <xsd:element name="off" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="ext" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chOff" type="CT_Point2D" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="chExt" type="CT_PositiveSize2D" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rot" type="ST_Angle" use="optional" default="0"/>
        <xsd:attribute name="flipH" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="flipV" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Point3D">
        <xsd:attribute name="x" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="y" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="z" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Vector3D">
        <xsd:attribute name="dx" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="dy" type="ST_Coordinate" use="required"/>
        <xsd:attribute name="dz" type="ST_Coordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SphereCoords">
        <xsd:attribute name="lat" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="lon" type="ST_PositiveFixedAngle" use="required"/>
        <xsd:attribute name="rev" type="ST_PositiveFixedAngle" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RelativeRect">
        <xsd:attribute name="l" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="t" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="r" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="b" type="ST_Percentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_RectAlignment">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tl"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="tr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="ctr"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="bl"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="br"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:group name="EG_ColorChoice">
        <xsd:choice>
            <xsd:element name="scrgbClr" type="CT_ScRgbColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="srgbClr" type="CT_SRgbColor" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="hslClr" type="CT_HslColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="sysClr" type="CT_SystemColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="schemeClr" type="CT_SchemeColor" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstClr" type="CT_PresetColor" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_Color">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorMRU">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BlackWhiteMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="clr"/>
            <xsd:enumeration value="auto"/>
            <xsd:enumeration value="gray"/>
            <xsd:enumeration value="ltGray"/>
            <xsd:enumeration value="invGray"/>
            <xsd:enumeration value="grayWhite"/>
            <xsd:enumeration value="blackGray"/>
            <xsd:enumeration value="blackWhite"/>
            <xsd:enumeration value="black"/>
            <xsd:enumeration value="white"/>
            <xsd:enumeration value="hidden"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Blob">
        <xsd:attribute ref="r:embed" use="optional" default=""/>
        <xsd:attribute ref="r:link" use="optional" default=""/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_EmbeddedWAVAudioFile">
        <xsd:attribute ref="r:embed" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_Hyperlink">
        <xsd:sequence>
            <xsd:element name="snd" type="CT_EmbeddedWAVAudioFile" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute ref="r:id" use="optional"/>
        <xsd:attribute name="invalidUrl" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="action" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="tgtFrame" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="tooltip" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="history" type="xsd:boolean" use="optional" default="true"/>
        <xsd:attribute name="highlightClick" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="endSnd" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:simpleType name="ST_DrawingElementId">
        <xsd:restriction base="xsd:unsignedInt"/>
    </xsd:simpleType>
    <xsd:attributeGroup name="AG_Locking">
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noRot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noEditPoints" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noAdjustHandles" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeArrowheads" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeShapeType" type="xsd:boolean" use="optional" default="false"/>
    </xsd:attributeGroup>
    <xsd:complexType name="CT_ConnectorLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ShapeLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
        <xsd:attribute name="noTextEdit" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_PictureLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
        <xsd:attribute name="noCrop" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noUngrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noRot" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectFrameLocking">
        <xsd:sequence>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="noGrp" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noDrilldown" type="xsd:boolean" use="optional" default="false"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="noSelect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noChangeAspect" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noMove" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="noResize" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ContentPartLocking">
        <xsd:sequence>
          <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Locking"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualDrawingProps">
        <xsd:sequence>
            <xsd:element name="hlinkClick" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="hlinkHover" type="CT_Hyperlink" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="ST_DrawingElementId" use="required"/>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="descr" type="xsd:string" use="optional" default=""/>
        <xsd:attribute name="hidden" type="xsd:boolean" use="optional" default="false"/>
        <xsd:attribute name="title" type="xsd:string" use="optional" default=""/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualDrawingShapeProps">
        <xsd:sequence>
            <xsd:element name="spLocks" type="CT_ShapeLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="txBox" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualConnectorProperties">
        <xsd:sequence>
            <xsd:element name="cxnSpLocks" type="CT_ConnectorLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="stCxn" type="CT_Connection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="endCxn" type="CT_Connection" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualPictureProperties">
        <xsd:sequence>
            <xsd:element name="picLocks" type="CT_PictureLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="preferRelativeResize" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualGroupDrawingShapeProps">
        <xsd:sequence>
            <xsd:element name="grpSpLocks" type="CT_GroupLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualGraphicFrameProperties">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:element name="graphicFrameLocks" type="CT_GraphicalObjectFrameLocking" minOccurs="0"
              maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_NonVisualContentPartProperties">
        <xsd:sequence>
            <xsd:element name="cpLocks" type="CT_ContentPartLocking" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="isComment" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObjectData">
        <xsd:sequence>
            <xsd:any minOccurs="0" maxOccurs="unbounded" processContents="strict"/>
        </xsd:sequence>
        <xsd:attribute name="uri" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GraphicalObject">
        <xsd:sequence>
            <xsd:element name="graphicData" type="CT_GraphicalObjectData"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="graphic" type="CT_GraphicalObject"/>
    <xsd:simpleType name="ST_ChartBuildStep">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="ptInCategory"/>
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="ptInSeries"/>
            <xsd:enumeration value="allPts"/>
            <xsd:enumeration value="gridLegend"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_DgmBuildStep">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sp"/>
            <xsd:enumeration value="bg"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationDgmElement">
        <xsd:attribute name="id" type="s:ST_Guid" use="optional" default="{00000000-0000-0000-0000-
          000000000000}"/>
        <xsd:attribute name="bldStep" type="ST_DgmBuildStep" use="optional" default="sp"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationChartElement">
        <xsd:attribute name="seriesIdx" type="xsd:int" use="optional" default="-1"/>
        <xsd:attribute name="categoryIdx" type="xsd:int" use="optional" default="-1"/>
        <xsd:attribute name="bldStep" type="ST_ChartBuildStep" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationElementChoice">
        <xsd:choice minOccurs="1" maxOccurs="1">
            <xsd:element name="dgm" type="CT_AnimationDgmElement"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:element name="chart" type="CT_AnimationChartElement"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:simpleType name="ST_AnimationBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="allAtOnce"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationDgmOnlyBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="one"/>
            <xsd:enumeration value="lvlOne"/>
            <xsd:enumeration value="lvlAtOnce"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationDgmBuildType">
        <xsd:union memberTypes="ST_AnimationBuildType ST_AnimationDgmOnlyBuildType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationDgmBuildProperties">
        <xsd:attribute name="bld" type="ST_AnimationDgmBuildType" use="optional" default="allAtOnce"/>
        <xsd:attribute name="rev" type="xsd:boolean" use="optional" default="false"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_AnimationChartOnlyBuildType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="series"/>
            <xsd:enumeration value="category"/>
            <xsd:enumeration value="seriesEl"/>
            <xsd:enumeration value="categoryEl"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_AnimationChartBuildType">
        <xsd:union memberTypes="ST_AnimationBuildType ST_AnimationChartOnlyBuildType"/>
    </xsd:simpleType>
    <xsd:complexType name="CT_AnimationChartBuildProperties">
        <xsd:attribute name="bld" type="ST_AnimationChartBuildType" use="optional"
          default="allAtOnce"/>
        <xsd:attribute name="animBg" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AnimationGraphicalObjectBuildProperties">
        <xsd:choice>
            <xsd:element name="bldDgm" type="CT_AnimationDgmBuildProperties"/>
            <xsd:element name="bldChart" type="CT_AnimationChartBuildProperties"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="CT_BackgroundFormatting">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_WholeE2oFormatting">
        <xsd:sequence>
            <xsd:element name="ln" type="CT_LineProperties" minOccurs="0" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:group ref="EG_EffectProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlUseShapeRectangle"/>
    <xsd:complexType name="CT_GvmlTextShape">
        <xsd:sequence>
            <xsd:element name="txBody" type="CT_TextBody" minOccurs="1" maxOccurs="1"/>
            <xsd:choice>
                <xsd:element name="useSpRect" type="CT_GvmlUseShapeRectangle" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="xfrm" type="CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvSpPr" type="CT_NonVisualDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlShape">
        <xsd:sequence>
            <xsd:element name="nvSpPr" type="CT_GvmlShapeNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="txSp" type="CT_GvmlTextShape" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlConnectorNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvCxnSpPr" type="CT_NonVisualConnectorProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlConnector">
        <xsd:sequence>
            <xsd:element name="nvCxnSpPr" type="CT_GvmlConnectorNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlPictureNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvPicPr" type="CT_NonVisualPictureProperties" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlPicture">
        <xsd:sequence>
            <xsd:element name="nvPicPr" type="CT_GvmlPictureNonVisual" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blipFill" type="CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="spPr" type="CT_ShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="style" type="CT_ShapeStyle" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGraphicFrameNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGraphicFramePr" type="CT_NonVisualGraphicFrameProperties"
              minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGraphicalObjectFrame">
        <xsd:sequence>
            <xsd:element name="nvGraphicFramePr" type="CT_GvmlGraphicFrameNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element ref="graphic" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="xfrm" type="CT_Transform2D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGroupShapeNonVisual">
        <xsd:sequence>
            <xsd:element name="cNvPr" type="CT_NonVisualDrawingProps" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="cNvGrpSpPr" type="CT_NonVisualGroupDrawingShapeProps" minOccurs="1"
              maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GvmlGroupShape">
        <xsd:sequence>
            <xsd:element name="nvGrpSpPr" type="CT_GvmlGroupShapeNonVisual" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="grpSpPr" type="CT_GroupShapeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="txSp" type="CT_GvmlTextShape"/>
                <xsd:element name="sp" type="CT_GvmlShape"/>
                <xsd:element name="cxnSp" type="CT_GvmlConnector"/>
                <xsd:element name="pic" type="CT_GvmlPicture"/>
                <xsd:element name="graphicFrame" type="CT_GvmlGraphicalObjectFrame"/>
                <xsd:element name="grpSp" type="CT_GvmlGroupShape"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetCameraType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyObliqueTopLeft"/>
            <xsd:enumeration value="legacyObliqueTop"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="legacyObliqueTopRight"/>
            <xsd:enumeration value="legacyObliqueLeft"/>
            <xsd:enumeration value="legacyObliqueFront"/>
            <xsd:enumeration value="legacyObliqueRight"/>
            <xsd:enumeration value="legacyObliqueBottomLeft"/>
            <xsd:enumeration value="legacyObliqueBottom"/>
            <xsd:enumeration value="legacyObliqueBottomRight"/>
            <xsd:enumeration value="legacyPerspectiveTopLeft"/>
            <xsd:enumeration value="legacyPerspectiveTop"/>
            <xsd:enumeration value="legacyPerspectiveTopRight"/>
            <xsd:enumeration value="legacyPerspectiveLeft"/>
            <xsd:enumeration value="legacyPerspectiveFront"/>
            <xsd:enumeration value="legacyPerspectiveRight"/>
            <xsd:enumeration value="legacyPerspectiveBottomLeft"/>
            <xsd:enumeration value="legacyPerspectiveBottom"/>
            <xsd:enumeration value="legacyPerspectiveBottomRight"/>
            <xsd:enumeration value="orthographicFront"/>
            <xsd:enumeration value="isometricTopUp"/>
            <xsd:enumeration value="isometricTopDown"/>
            <xsd:enumeration value="isometricBottomUp"/>
            <xsd:enumeration value="isometricBottomDown"/>
            <xsd:enumeration value="isometricLeftUp"/>
            <xsd:enumeration value="isometricLeftDown"/>
            <xsd:enumeration value="isometricRightUp"/>
            <xsd:enumeration value="isometricRightDown"/>
            <xsd:enumeration value="isometricOffAxis1Left"/>
            <xsd:enumeration value="isometricOffAxis1Right"/>
            <xsd:enumeration value="isometricOffAxis1Top"/>
            <xsd:enumeration value="isometricOffAxis2Left"/>
            <xsd:enumeration value="isometricOffAxis2Right"/>
            <xsd:enumeration value="isometricOffAxis2Top"/>
            <xsd:enumeration value="isometricOffAxis3Left"/>
            <xsd:enumeration value="isometricOffAxis3Right"/>
            <xsd:enumeration value="isometricOffAxis3Bottom"/>
            <xsd:enumeration value="isometricOffAxis4Left"/>
            <xsd:enumeration value="isometricOffAxis4Right"/>
            <xsd:enumeration value="isometricOffAxis4Bottom"/>
            <xsd:enumeration value="obliqueTopLeft"/>
            <xsd:enumeration value="obliqueTop"/>
            <xsd:enumeration value="obliqueTopRight"/>
            <xsd:enumeration value="obliqueLeft"/>
            <xsd:enumeration value="obliqueRight"/>
            <xsd:enumeration value="obliqueBottomLeft"/>
            <xsd:enumeration value="obliqueBottom"/>
            <xsd:enumeration value="obliqueBottomRight"/>
            <xsd:enumeration value="perspectiveFront"/>
            <xsd:enumeration value="perspectiveLeft"/>
            <xsd:enumeration value="perspectiveRight"/>
            <xsd:enumeration value="perspectiveAbove"/>
            <xsd:enumeration value="perspectiveBelow"/>
            <xsd:enumeration value="perspectiveAboveLeftFacing"/>
            <xsd:enumeration value="perspectiveAboveRightFacing"/>
            <xsd:enumeration value="perspectiveContrastingLeftFacing"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="perspectiveContrastingRightFacing"/>
            <xsd:enumeration value="perspectiveHeroicLeftFacing"/>
            <xsd:enumeration value="perspectiveHeroicRightFacing"/>
            <xsd:enumeration value="perspectiveHeroicExtremeLeftFacing"/>
            <xsd:enumeration value="perspectiveHeroicExtremeRightFacing"/>
            <xsd:enumeration value="perspectiveRelaxed"/>
            <xsd:enumeration value="perspectiveRelaxedModerately"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_FOVAngle">
        <xsd:restriction base="ST_Angle">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="10800000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Camera">
        <xsd:sequence>
            <xsd:element name="rot" type="CT_SphereCoords" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetCameraType" use="required"/>
        <xsd:attribute name="fov" type="ST_FOVAngle" use="optional"/>
        <xsd:attribute name="zoom" type="ST_PositivePercentage" use="optional" default="100%"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_LightRigDirection">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="tl"/>
            <xsd:enumeration value="t"/>
            <xsd:enumeration value="tr"/>
            <xsd:enumeration value="l"/>
            <xsd:enumeration value="r"/>
            <xsd:enumeration value="bl"/>
            <xsd:enumeration value="b"/>
            <xsd:enumeration value="br"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="ST_LightRigType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyFlat1"/>
            <xsd:enumeration value="legacyFlat2"/>
            <xsd:enumeration value="legacyFlat3"/>
            <xsd:enumeration value="legacyFlat4"/>
            <xsd:enumeration value="legacyNormal1"/>
            <xsd:enumeration value="legacyNormal2"/>
            <xsd:enumeration value="legacyNormal3"/>
            <xsd:enumeration value="legacyNormal4"/>
            <xsd:enumeration value="legacyHarsh1"/>
            <xsd:enumeration value="legacyHarsh2"/>
            <xsd:enumeration value="legacyHarsh3"/>
            <xsd:enumeration value="legacyHarsh4"/>
            <xsd:enumeration value="threePt"/>
            <xsd:enumeration value="balanced"/>
            <xsd:enumeration value="soft"/>
            <xsd:enumeration value="harsh"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:enumeration value="flood"/>
            <xsd:enumeration value="contrasting"/>
            <xsd:enumeration value="morning"/>
            <xsd:enumeration value="sunrise"/>
            <xsd:enumeration value="sunset"/>
            <xsd:enumeration value="chilly"/>
            <xsd:enumeration value="freezing"/>
            <xsd:enumeration value="flat"/>
            <xsd:enumeration value="twoPt"/>
            <xsd:enumeration value="glow"/>
            <xsd:enumeration value="brightRoom"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_LightRig">
        <xsd:sequence>
            <xsd:element name="rot" type="CT_SphereCoords" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rig" type="ST_LightRigType" use="required"/>
        <xsd:attribute name="dir" type="ST_LightRigDirection" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_Scene3D">
        <xsd:sequence>
            <xsd:element name="camera" type="CT_Camera" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lightRig" type="CT_LightRig" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="backdrop" type="CT_Backdrop" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_Backdrop">
        <xsd:sequence>
            <xsd:element name="anchor" type="CT_Point3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="norm" type="CT_Vector3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="up" type="CT_Vector3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BevelPresetType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="relaxedInset"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="slope"/>
            <xsd:enumeration value="cross"/>
            <xsd:enumeration value="angle"/>
            <xsd:enumeration value="softRound"/>
            <xsd:enumeration value="convex"/>
            <xsd:enumeration value="coolSlant"/>
            <xsd:enumeration value="divot"/>
            <xsd:enumeration value="riblet"/>
            <xsd:enumeration value="hardEdge"/>
            <xsd:enumeration value="artDeco"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Bevel">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="w" type="ST_PositiveCoordinate" use="optional" default="76200"/>
        <xsd:attribute name="h" type="ST_PositiveCoordinate" use="optional" default="76200"/>
        <xsd:attribute name="prst" type="ST_BevelPresetType" use="optional" default="circle"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetMaterialType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="legacyMatte"/>
            <xsd:enumeration value="legacyPlastic"/>
            <xsd:enumeration value="legacyMetal"/>
            <xsd:enumeration value="legacyWireframe"/>
            <xsd:enumeration value="matte"/>
            <xsd:enumeration value="plastic"/>
            <xsd:enumeration value="metal"/>
            <xsd:enumeration value="warmMatte"/>
            <xsd:enumeration value="translucentPowder"/>
            <xsd:enumeration value="powder"/>
            <xsd:enumeration value="dkEdge"/>
            <xsd:enumeration value="softEdge"/>
            <xsd:enumeration value="clear"/>
            <xsd:enumeration value="flat"/>
            <xsd:enumeration value="softmetal"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Shape3D">
        <xsd:sequence>
            <xsd:element name="bevelT" type="CT_Bevel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bevelB" type="CT_Bevel" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extrusionClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="contourClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="z" type="ST_Coordinate" use="optional" default="0"/>
        <xsd:attribute name="extrusionH" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="contourW" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="prstMaterial" type="ST_PresetMaterialType" use="optional"
          default="warmMatte"/>
    </xsd:complexType>
    <xsd:complexType name="CT_FlatText">
        <xsd:attribute name="z" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:group name="EG_Text3D">
        <xsd:choice>
            <xsd:element name="sp3d" type="CT_Shape3D" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="flatTx" type="CT_FlatText" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_AlphaBiLevelEffect">
        <xsd:attribute name="thresh" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaCeilingEffect"/>
    <xsd:complexType name="CT_AlphaFloorEffect"/>
    <xsd:complexType name="CT_AlphaInverseEffect">
        <xsd:sequence>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaModulateFixedEffect">
        <xsd:attribute name="amt" type="ST_PositivePercentage" use="optional" default="100%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaOutsetEffect">
        <xsd:attribute name="rad" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaReplaceEffect">
        <xsd:attribute name="a" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BiLevelEffect">
        <xsd:attribute name="thresh" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BlurEffect">
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="grow" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorChangeEffect">
        <xsd:sequence>
            <xsd:element name="clrFrom" type="CT_Color" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrTo" type="CT_Color" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="useA" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_ColorReplaceEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_DuotoneEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="2" maxOccurs="2"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GlowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GrayscaleEffect"/>
    <xsd:complexType name="CT_HSLEffect">
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="sat" type="ST_FixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="lum" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_InnerShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_LuminanceEffect">
        <xsd:attribute name="bright" type="ST_FixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="contrast" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_OuterShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional" default="b"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetShadowVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="shdw1"/>
            <xsd:enumeration value="shdw2"/>
            <xsd:enumeration value="shdw3"/>
            <xsd:enumeration value="shdw4"/>
            <xsd:enumeration value="shdw5"/>
            <xsd:enumeration value="shdw6"/>
            <xsd:enumeration value="shdw7"/>
            <xsd:enumeration value="shdw8"/>
            <xsd:enumeration value="shdw9"/>
            <xsd:enumeration value="shdw10"/>
            <xsd:enumeration value="shdw11"/>
            <xsd:enumeration value="shdw12"/>
            <xsd:enumeration value="shdw13"/>
            <xsd:enumeration value="shdw14"/>
            <xsd:enumeration value="shdw15"/>
            <xsd:enumeration value="shdw16"/>
            <xsd:enumeration value="shdw17"/>
            <xsd:enumeration value="shdw18"/>
            <xsd:enumeration value="shdw19"/>
            <xsd:enumeration value="shdw20"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PresetShadowEffect">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetShadowVal" use="required"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
    </xsd:complexType>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
    <xsd:complexType name="CT_ReflectionEffect">
        <xsd:attribute name="blurRad" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="stA" type="ST_PositiveFixedPercentage" use="optional" default="100%"/>
        <xsd:attribute name="stPos" type="ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="endA" type="ST_PositiveFixedPercentage" use="optional" default="0%"/>
        <xsd:attribute name="endPos" type="ST_PositiveFixedPercentage" use="optional" default="100%"/>
        <xsd:attribute name="dist" type="ST_PositiveCoordinate" use="optional" default="0"/>
        <xsd:attribute name="dir" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="fadeDir" type="ST_PositiveFixedAngle" use="optional" default="5400000"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional" default="b"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional" default="true"/>
    </xsd:complexType>
    <xsd:complexType name="CT_RelativeOffsetEffect">
        <xsd:attribute name="tx" type="ST_Percentage" use="optional" default="0%"/>
        <xsd:attribute name="ty" type="ST_Percentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_SoftEdgesEffect">
        <xsd:attribute name="rad" type="ST_PositiveCoordinate" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TintEffect">
        <xsd:attribute name="hue" type="ST_PositiveFixedAngle" use="optional" default="0"/>
        <xsd:attribute name="amt" type="ST_FixedPercentage" use="optional" default="0%"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TransformEffect">
        <xsd:attribute name="sx" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional" default="100%"/>
        <xsd:attribute name="kx" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="ky" type="ST_FixedAngle" use="optional" default="0"/>
        <xsd:attribute name="tx" type="ST_Coordinate" use="optional" default="0"/>
        <xsd:attribute name="ty" type="ST_Coordinate" use="optional" default="0"/>
    </xsd:complexType>
    <xsd:complexType name="CT_NoFillProperties"/>
    <xsd:complexType name="CT_SolidColorFillProperties">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_LinearShadeProperties">
        <xsd:attribute name="ang" type="ST_PositiveFixedAngle" use="optional"/>
        <xsd:attribute name="scaled" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PathShadeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="shape"/>
            <xsd:enumeration value="circle"/>
            <xsd:enumeration value="rect"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PathShadeProperties">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:sequence>
            <xsd:element name="fillToRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="path" type="ST_PathShadeType" use="optional"/>
    </xsd:complexType>
    <xsd:group name="EG_ShadeProperties">
        <xsd:choice>
            <xsd:element name="lin" type="CT_LinearShadeProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="path" type="CT_PathShadeProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_TileFlipMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="none"/>
            <xsd:enumeration value="x"/>
            <xsd:enumeration value="y"/>
            <xsd:enumeration value="xy"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_GradientStop">
        <xsd:sequence>
            <xsd:group ref="EG_ColorChoice" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="pos" type="ST_PositiveFixedPercentage" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GradientStopList">
        <xsd:sequence>
            <xsd:element name="gs" type="CT_GradientStop" minOccurs="2" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_GradientFillProperties">
        <xsd:sequence>
            <xsd:element name="gsLst" type="CT_GradientStopList" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_ShadeProperties" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="tileRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="flip" type="ST_TileFlipMode" use="optional"/>
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_TileInfoProperties">
        <xsd:attribute name="tx" type="ST_Coordinate" use="optional"/>
        <xsd:attribute name="ty" type="ST_Coordinate" use="optional"/>
        <xsd:attribute name="sx" type="ST_Percentage" use="optional"/>
        <xsd:attribute name="sy" type="ST_Percentage" use="optional"/>
        <xsd:attribute name="flip" type="ST_TileFlipMode" use="optional"/>
        <xsd:attribute name="algn" type="ST_RectAlignment" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_StretchInfoProperties">
        <xsd:sequence>
            <xsd:element name="fillRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_FillModeProperties">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
        <xsd:choice>
            <xsd:element name="tile" type="CT_TileInfoProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="stretch" type="CT_StretchInfoProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_BlipCompression">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="email"/>
            <xsd:enumeration value="screen"/>
            <xsd:enumeration value="print"/>
            <xsd:enumeration value="hqprint"/>
            <xsd:enumeration value="none"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_Blip">
        <xsd:sequence>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="alphaBiLevel" type="CT_AlphaBiLevelEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaCeiling" type="CT_AlphaCeilingEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaFloor" type="CT_AlphaFloorEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="alphaInv" type="CT_AlphaInverseEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="alphaMod" type="CT_AlphaModulateEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaModFix" type="CT_AlphaModulateFixedEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="alphaRepl" type="CT_AlphaReplaceEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="biLevel" type="CT_BiLevelEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="blur" type="CT_BlurEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="clrChange" type="CT_ColorChangeEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="clrRepl" type="CT_ColorReplaceEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="duotone" type="CT_DuotoneEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="1"
                  maxOccurs="1"/>
                <xsd:element name="grayscl" type="CT_GrayscaleEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="hsl" type="CT_HSLEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="lum" type="CT_LuminanceEffect" minOccurs="1" maxOccurs="1"/>
                <xsd:element name="tint" type="CT_TintEffect" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="AG_Blob"/>
        <xsd:attribute name="cstate" type="ST_BlipCompression" use="optional" default="none"/>
    </xsd:complexType>
    <xsd:complexType name="CT_BlipFillProperties">
        <xsd:sequence>
            <xsd:element name="blip" type="CT_Blip" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="srcRect" type="CT_RelativeRect" minOccurs="0" maxOccurs="1"/>
            <xsd:group ref="EG_FillModeProperties" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="dpi" type="xsd:unsignedInt" use="optional"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="rotWithShape" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:simpleType name="ST_PresetPatternVal">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="pct5"/>
            <xsd:enumeration value="pct10"/>
            <xsd:enumeration value="pct20"/>
            <xsd:enumeration value="pct25"/>
            <xsd:enumeration value="pct30"/>
            <xsd:enumeration value="pct40"/>
            <xsd:enumeration value="pct50"/>
            <xsd:enumeration value="pct60"/>
            <xsd:enumeration value="pct70"/>
            <xsd:enumeration value="pct75"/>
            <xsd:enumeration value="pct80"/>
            <xsd:enumeration value="pct90"/>
            <xsd:enumeration value="horz"/>
            <xsd:enumeration value="vert"/>
            <xsd:enumeration value="ltHorz"/>
            <xsd:enumeration value="ltVert"/>
            <xsd:enumeration value="dkHorz"/>
            <xsd:enumeration value="dkVert"/>
            <xsd:enumeration value="narHorz"/>
            <xsd:enumeration value="narVert"/>
            <xsd:enumeration value="dashHorz"/>
            <xsd:enumeration value="dashVert"/>
            <xsd:enumeration value="cross"/>
            <xsd:enumeration value="dnDiag"/>
            <xsd:enumeration value="upDiag"/>
            <xsd:enumeration value="ltDnDiag"/>
            <xsd:enumeration value="ltUpDiag"/>
            <xsd:enumeration value="dkDnDiag"/>
            <xsd:enumeration value="dkUpDiag"/>
            <xsd:enumeration value="wdDnDiag"/>
            <xsd:enumeration value="wdUpDiag"/>
            <xsd:enumeration value="dashDnDiag"/>
            <xsd:enumeration value="dashUpDiag"/>
            <xsd:enumeration value="diagCross"/>
            <xsd:enumeration value="smCheck"/>
            <xsd:enumeration value="lgCheck"/>
            <xsd:enumeration value="smGrid"/>
            <xsd:enumeration value="lgGrid"/>
            <xsd:enumeration value="dotGrid"/>
            <xsd:enumeration value="smConfetti"/>
            <xsd:enumeration value="lgConfetti"/>
            <xsd:enumeration value="horzBrick"/>
            <xsd:enumeration value="diagBrick"/>
            <xsd:enumeration value="solidDmnd"/>
            <xsd:enumeration value="openDmnd"/>
            <xsd:enumeration value="dotDmnd"/>
            <xsd:enumeration value="plaid"/>
            <xsd:enumeration value="sphere"/>
            <xsd:enumeration value="weave"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved

```
            <xsd:enumeration value="divot"/>
            <xsd:enumeration value="shingle"/>
            <xsd:enumeration value="wave"/>
            <xsd:enumeration value="trellis"/>
            <xsd:enumeration value="zigZag"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_PatternFillProperties">
        <xsd:sequence>
            <xsd:element name="fgClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="bgClr" type="CT_Color" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="prst" type="ST_PresetPatternVal" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_GroupFillProperties"/>
    <xsd:group name="EG_FillProperties">
        <xsd:choice>
            <xsd:element name="noFill" type="CT_NoFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="solidFill" type="CT_SolidColorFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="gradFill" type="CT_GradientFillProperties" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="blipFill" type="CT_BlipFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="pattFill" type="CT_PatternFillProperties" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grpFill" type="CT_GroupFillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_FillProperties">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_FillEffect">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="ST_BlendMode">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="over"/>
            <xsd:enumeration value="mult"/>
            <xsd:enumeration value="screen"/>
            <xsd:enumeration value="darken"/>
            <xsd:enumeration value="lighten"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_FillOverlayEffect">
        <xsd:sequence>
            <xsd:group ref="EG_FillProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blend" type="ST_BlendMode" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectReference">
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

 ©ISO/IEC 2016 – All rights reserved

```
        <xsd:attribute name="ref" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:group name="EG_Effect">
        <xsd:choice>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effect" type="CT_EffectReference" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaBiLevel" type="CT_AlphaBiLevelEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaCeiling" type="CT_AlphaCeilingEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaFloor" type="CT_AlphaFloorEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaInv" type="CT_AlphaInverseEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaMod" type="CT_AlphaModulateEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaModFix" type="CT_AlphaModulateFixedEffect" minOccurs="1"
              maxOccurs="1"/>
            <xsd:element name="alphaOutset" type="CT_AlphaOutsetEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="alphaRepl" type="CT_AlphaReplaceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="biLevel" type="CT_BiLevelEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blend" type="CT_BlendEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="blur" type="CT_BlurEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrChange" type="CT_ColorChangeEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="clrRepl" type="CT_ColorReplaceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="duotone" type="CT_DuotoneEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fill" type="CT_FillEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="glow" type="CT_GlowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="grayscl" type="CT_GrayscaleEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="hsl" type="CT_HSLEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="innerShdw" type="CT_InnerShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="lum" type="CT_LuminanceEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="outerShdw" type="CT_OuterShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="prstShdw" type="CT_PresetShadowEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="reflection" type="CT_ReflectionEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="relOff" type="CT_RelativeOffsetEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="softEdge" type="CT_SoftEdgesEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="tint" type="CT_TintEffect" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="xfrm" type="CT_TransformEffect" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:simpleType name="ST_EffectContainerType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="sib"/>
            <xsd:enumeration value="tree"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="CT_EffectContainer">
        <xsd:group ref="EG_Effect" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:attribute name="type" type="ST_EffectContainerType" use="optional" default="sib"/>
        <xsd:attribute name="name" type="xsd:token" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="CT_AlphaModulateEffect">
        <xsd:sequence>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="CT_BlendEffect">
        <xsd:sequence>
            <xsd:element name="cont" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="blend" type="ST_BlendMode" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="CT_EffectList">
        <xsd:sequence>
            <xsd:element name="blur" type="CT_BlurEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="fillOverlay" type="CT_FillOverlayEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="glow" type="CT_GlowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="innerShdw" type="CT_InnerShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="outerShdw" type="CT_OuterShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="prstShdw" type="CT_PresetShadowEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="reflection" type="CT_ReflectionEffect" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="softEdge" type="CT_SoftEdgesEffect" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:group name="EG_EffectProperties">
        <xsd:choice>
            <xsd:element name="effectLst" type="CT_EffectList" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="effectDag" type="CT_EffectContainer" minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="CT_EffectProperties">
        <xsd:sequence>
            <xsd:group ref="EG_EffectProperties" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="blip" type="CT_Blip"/>
    <xsd:simpleType name="ST_ShapeType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="line"/>
            <xsd:enumeration value="lineInv"/>
            <xsd:enumeration value="triangle"/>
            <xsd:enumeration value="rtTriangle"/>
            <xsd:enumeration value="rect"/>
            <xsd:enumeration value="diamond"/>
            <xsd:enumeration value="parallelogram"/>
            <xsd:enumeration value="trapezoid"/>
            <xsd:enumeration value="nonIsoscelesTrapezoid"/>
            <xsd:enumeration value="pentagon"/>
            <xsd:enumeration value="hexagon"/>
            <xsd:enumeration value="heptagon"/>
            <xsd:enumeration value="octagon"/>
            <xsd:enumeration value="decagon"/>
            <xsd:enumeration value="dodecagon"/>
            <xsd:enumeration value="star4"/>
            <xsd:enumeration value="star5"/>
            <xsd:enumeration value="star6"/>
            <xsd:enumeration value="star7"/>
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 29500-1:2016

©ISO/IEC 2016 – All rights reserved