



INTERNATIONAL STANDARD ISO/IEC 26300:2006
TECHNICAL CORRIGENDUM 2

Published 2011-12-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Open Document Format for Office Applications (OpenDocument) v1.0

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Format de document ouvert pour applications de bureau (OpenDocument) v1.0

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to ISO/IEC 26300:2006 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*. The text is technically equivalent to Part 2 of 'Open Document Format for Office Applications (OpenDocument) Version 1.0 Errata' published by OASIS.

NOTE Some changes made to ODF Version 1.0 by Part 2 of the Errata published by OASIS have been omitted due to conflict with changes made by OASIS in preparing ODF Version 1.1. A project to align ISO/IEC 26300:2006 with ODF Version 1.1, by Amendment of the International Standard, is currently in progress. It is considered inappropriate to make changes in this Technical Corrigendum that would have to be reversed or changed further by an Amendment.

This Technical Corrigendum should be read in conjunction with ISO/IEC 26300:2006 and the associated Technical Corrigendum 1. The current edition of ISO/IEC 26300 should be understood by first applying the changes specified in Technical Corrigendum 1, then the changes specified in this Technical Corrigendum.

Clause 1.3, "Table 1", column 1, page 31, line 9

Replace "drawing" with "draw".

Clause 1.6, page 34, line 12

Replace the entire text of Clause 1.6 with:

“1.6 White-Space Processing

ODF processing of whitespace characters is in conformance with the provisions of [XML 1.0].

In addition, ODF processors shall ignore all element children ([RNG] section 5, Data Model) of ODF-defined elements that are strings consisting entirely of whitespace characters and which do not satisfy a pattern of the ODF schema definition for the element.

Any special treatment of additional occurrences of whitespace characters depends on the specific definitions of individual ODF elements, attributes, and their datatypes. See, in particular, section 5.1.1.”

Clause 2.1.2, “Version”, page 38, line 44

Delete:

“If the file has a version known to an XML processor, it may validate the document. Otherwise, it is optional to validate the document, but the document must be well formed.”

Clause 3.1.15, page 61, line 32

Replace:

“The manner in which the language is represented is similar to the language tag described in [RFC3066]. It consists of a two or three letter Language Code taken from the ISO 639 standard optionally followed by a hyphen (-) and a two-letter Country Code taken from the ISO 3166 standard.”

with:

“The syntax and semantics of the language tag are specified in [RFC3066].”

Clause 4.4.1, “Protected Sections”, page 75, line 36

After:

“To avoid saving the password directly into the XML file, only a hash value of the password is stored.”

insert:

“The hashing is implementation-dependent.”

Clause 6.3, page 107, line 36

Replace:

“OpenDocument text documents can contain variables, which are processed or displayed using variable fields.”

with:

“OpenDocument documents can contain variables, which are processed or displayed using variable fields.”

Clause 6.3, page 107, line 40

Delete:

“In the OpenDocument file format, a variable must be declared at the beginning of a document”.

Clause 6.7.2, page 140, line 31

Replace:

“If the value of a field is fixed, the value of the field element to which this attribute is attached is preserved in all future edits of the document. If the value of the field is not fixed, the value of the field may be replaced by a new value when the document is edited.”

with:

“The defined values of the `text:fixed` attribute are:

- `true`: value of the field element where this attribute appears is preserved.
- `false`: value of the field element where this attribute appears may be changed.”

Clause 7.3.1, “Use Outline”, page 153, line 19

Delete:

“See section 7.1 for more information on index marks.”

Clause 7.3.1, “Index Scope”, page 154, line 6

Replace:

“The `text:index-scope` attribute determines whether the table-of-content is generated for the whole document, or only for the current chapter.”

with:

“The `text:index-scope` attribute specifies whether index entries from an entire document or from the chapter that contains the `<text:table-of-content>` element are used to construct a table of contents. The default value and the determination of the chapter are implementation-dependent.”

Clause 7.3.2, page 154, line 33

Delete:

“(See below.)”

Clause 7.4.1, “Use Caption”, page 157, line 4

Replace:

“Each object contained in a text document has a name. In addition, images also have a caption. The image caption or the image name can be gathered for the index of illustrations.”

with:

“The `text:use-caption` attribute specifies whether the captions or names of illustrations are used for an index.

The defined values of the `text:use-caption` attribute are:

- `true`: illustration captions are used for an index.
- `false`: illustration names are used for an index.”

Clause 7.7.1, “Copy Outline Levels”, page 163, line 15

Replace:

“This attribute can have a value of true or false.

If the value is true, the entries are gathered at the outline level of the source element to which they refer.

If the value is false, all index entries gathered are at the top outline level. For example, if an image appears in

section 1.2.3, the entry for the image is located at outline level 3.”

with:

“The `text:copy-outline-levels` attribute specifies whether index entries are indented according to the outline level of their source.

The defined values for the `text:copy-outline-levels` attribute are:

- `false`: no indentation is added.
- `true`: index entries are indented according to the outline level of their source.”

Clause 7.8.2, page 168, line 23

Insert before the schema fragment the following new paragraph:

“The format of an index mark file is implementation-dependent.”

Clause 7.12, page 171, line 37

Replace:

“There are eight types of index entries, as follows:”

with

“There are seven types of index entry, as follows:”

Clause 8.1.1, “Protected”, page 180, line 44

Replace:

“If a table is protected, all of the table elements and the cell elements with a `style:cell-protect` attribute set to `true` are protected.”

with

“If a table is protected, all of the table's cell elements with a `style:cell-protect` attribute set to a different value than `none` are protected.”

Clause 8.1.1, page 181, line 3

Insert before the schema fragment the following new paragraph:

“The hashing is implementation-dependent.”

Clause 8.1.2, “Visibility”, page 182, line 44

Replace the entire paragraph with the following text:

“The `table:visibility` attribute specifies the conditions under which the table row is visible. The values of the attribute are:

- `visible`: the row is visible
- `collapse`: the row is not visible
- `filter`: the row is not visible as the result of applying a filter (see section 8.7)

The default is `visible`.”

Clause 8.1.3, “Value Type”, page 187, line 27

Replace:

“The `table:value-type` attribute specifies the type of value that can appear in a cell.”

with

“The `office:value-type` attribute specifies the type of value that can appear in a cell.”

Clause 8.1.3, “Cell Current Numeric Value”, page 187, line 40

Replace:

“This attribute is only evaluated for cells that contain the following data types:”

with

“This attribute is present only whenever there is an `office:value-type` attribute with one of the following values:”

Clause 8.1.3, “Cell Current Currency”, page 188, line 4

Replace:

“The `tableoffice:currency` attribute specifies the current currency value of a cell. The value of this attribute is usually currency information such as DEM or EUR. This attribute is only evaluated for cells whose data type is `currency`.”

with

“The `office:currency` attribute specifies the currency system in which the value of the `office:value` attribute is expressed. The value of this attribute is usually a currency identifier, such as “DEM” or “EUR”. This attribute may be present only when the `office:value-type` attribute value is “`currency`”.”

Clause 8.1.3, “Cell Current Date Value”, page 188, line 8

Replace:

“This attribute is only evaluated for cells whose data type is `date`.”

with

“This attribute is present only whenever the `office:value-type` attribute value is “`date`”.”

Clause 8.1.3, “Cell Current Time Value”, page 188, line 12

Replace:

“This attribute is only evaluated for cells whose data type is `time`.”

with

“This attribute is present only whenever the `office:value-type` attribute is “`time`”.”

Clause 8.1.3, “Cell Current Boolean Value”, page 188, line 15

Replace:

“This attribute is only evaluated for cells whose data type is `boolean`.”

with

“This attribute is present only whenever the `office:value-type` attribute value is “`boolean`”.”

Clause 8.1.3, “Cell Current String Value”, page 188, line 18

Replace:

“This attribute is only evaluated for cells whose data type is `string`.”

with

“This attribute may be present only whenever the `office:value-type` attribute value is “`string`”.”

Clause 8.1.3, “Table Cell Protection”, page 188, line 21

Replace:

“The `table:protected` attribute protects the table cells.”

with

“The `table:protect` attribute protects the table cells.”

Clause 8.1.3, “Table Cell Protection”, page 188, line 30

Replace:

“This attribute is not related to the `table:protected` attribute for table elements (see section 8.1.1) and the `table:cell-protect` attribute for table cell styles (see section 15.11.14).”

with:

“This attribute is not related to the `table:protected` attribute for table elements (see section 8.1.1) and the `style:cell-protect` attribute for table cell styles (see section 15.11.14).”

Clause 8.3.4, page 202, line 13

Replace:

“The `<table:shapes>` element contains all graphic shapes with an anchor on the table this element is a child of.”

with

“The `<table:shapes>` element contains all the elements that represent graphic shapes (see section 9.2) that are anchored on a table where this element occurs.”

Clause 8.4.3, page 203, line 25

Replace:

“The `<table:detective>` element has two purposes. One the one hand, it contains information about relations that exist between table cells because of formulas and that should be highlighted in the UI. On the other hand, the element contains information about cells that are highlighted currently in the UI either because of the relations mentioned above or because of error conditions.”

with

“The `<table:detective>` element contains information about what relationships between the current cell and other cells are revealed in the presentation of the table.”

Clause 8.4.4, page 204, line 2

Replace:

“The `<table:operation>` element specifies that certain relations that exist between the cell the element is a child of and other cells should be made visible or invisible in the UI. One and the same detective operation can be applied multiple times to the same cell. In this case, the second operation is applied to the resulting cells of the first operation and so on. This means that an operation not necessarily is applied to the cell the operation is defined in, but also to other cells, and that it therefor can interact with operations defined in other cells. This especially applies to operations that make relations invisible. To get a determinate behavior, operations have an index and are applied in the order of that index. The attributes associated with the `<table:operation>` element are:”

with

“The `<table:operation>` element specifies both the type of detective operation that leads to the discovery of relationships between cells (`table:name` attribute) and the order which those operations are applied (`table:index` attribute). Once relationships between cells have been discovered, those cells are highlighted

to show those relationships. The attributes associated with the `<table:operation>` element are:”

Clause 8.4.4, “Name”, page 204, line 20

Replace

“The `table:name` attribute specifies the name of the detective operation. Possible names are `trace-dependents`, `remove-dependents`, `trace-precedents`, `remove-precedents` and `trace-errors`. `trace-dependents` and `remove-dependents` displays or hides cells that use the value of the current cell in their formula. `Trace-precedents` and `remove-precedents` displays or hides cells whose value is used in the formula of the current cell. `Trace-errors` displays cells that cause an error while calculating the result of the current cell's formula.”

with

“The `table:name` attribute specifies the name of a detective operation.

The defined values for the `table:name` attribute are:

- `remove-dependents`: removes highlighting from cells that use the value of the current cell in their formula.
- `remove-precedents`: removes highlighting from cells whose values are use in the formula of the current cell.
- `trace-dependents`: highlights cells that use the value of the current cell in their formula.
- `trace-errors`: highlights cells that cause an error while calculating the result of the current cell's formula.
- `trace-precedents`: highlights cells whose values are use in the formula of the current cell.

The nature of the highlighting imposed or removed from cells as the result of detective operations is implementation dependent.”

Clause 8.5.1, page 206, line 24

At the end of the paragraph insert:

“The hashing is implementation-dependent.”

Clause 8.5.2, “Search Criteria Must Apply to Whole Cell”, page 207, line 39

Replace:

“The `table:search-criteria-must-apply-to-whole-cell` attribute specifies whether or not the specified search criteria, according to the regular expression used, must apply to the entire cell contents.”

with

“The `table:search-criteria-must-apply-to-whole-cell` attribute specifies whether a search pattern from the `table:value` attribute matches the entire content of a cell. The defined values of the `table:search-criteria-must-apply-to-whole-cell` are:

- `false`: a search pattern can match any substring at any position in a cell.
- `true`: a search pattern must match the entire content of a cell.

The `table:search-criteria-must-apply-to-whole-cell` attribute is only used with the `<table:filter-condition>` element when its `table:data-type` attribute has the value `text` and its `table:operator` attribute has one of the values:

- `match` (i.e. matches)
- `!match` (i.e. does not match)
- `=` (i.e. equal to)

- != (i.e. not equal to).”

Clause 8.5.2, “Use Regular Expressions”, page 208, line 21

Insert before the schema fragment the following new paragraph:

“The syntax and semantics of the regular expressions are implementation-dependent.”

Clause 8.5.3, “Condition”, page 210, line 42

Replace:

“The value of this attribute should be a namespace prefix, followed by a Boolean expression.

A typical syntax of the expression may be similar to the XPath syntax. The following are valid conditions:

- `Condition ::= ExtendedTrueCondition | TrueFunction 'and' TrueCondition`
- `TrueFunction ::= cell-content-is-whole-number() | cell-content-is-decimal-number() | cell-content-is-date() | cell-content-is-time() | cell-content-is-text()`
- `ExtendedTrueCondition ::= ExtendedGetFunction | cell-content-text-length() Operator Value`
- `TrueCondition ::= GetFunction | cell-content() Operator Value`
- `GetFunction ::= cell-content-is-between(Value, Value) | cell-content-is-not-between(Value, Value)`
- `ExtendedGetFunction ::= cell-content-text-length-is-between(Value, Value) | cell-content-text-length-is-not-between(Value, Value) | cell-content-is-in-list(StringList)`
- `Operator ::= '<' | '>' | '<=' | '>=' | '=' | '!='`
- `Value ::= NumberValue | String | Formula`
- `StringList ::= StringList ';' String | String`
- A `Formula` is a formula without an equals (=) sign at the beginning. See section for more information.
- A `String` comprises one or more characters surrounded by quotation marks.
- A `NumberValue` is a whole or decimal number. It must not contain comma separators for number of 1000 or greater.”

with:

“The value of this attribute should be a namespace prefix, followed by an implementation-dependent Boolean expression.”

Clause 8.6.10, “Function”, page 227, line 36

Insert before the schema fragment the following new paragraph:

“These functions are implementation-dependent.”

Clause 8.7.1, “Condition Source Range Address”, page 229, line 32

In the example replace column headings “F G G I” with “F G H I”.

Clause 8.8, page 232, line 43

Replace:

“Data pilot tables allow it to analyze and evaluate data contained in spreadsheet tables.”

with:

“Data pilot tables enable users to analyze and evaluate data extracted from a range of structured data sources, such as spreadsheet tables, database tables, or external service components.”

Clause 8.8.4, “Function”, page 241, line 14

Insert before the schema fragment the following new paragraph:

“These functions are implementation-dependent.”

Clause 8.8.4, “Used Hierarchy”, page 241, line 43

Replace:

“The value means that no hierarchy should be applied at all.”

with:

“The value `-1` means no hierarchy should be applied at all.”

Clause 8.8.7, “Function”, page 243, line 16

Insert before the schema fragment the following new paragraph:

“These functions are implementation-dependent.”

Clause 8.9, “Function”, page 253, line 1

Insert before the schema fragment the following new paragraph:

“These functions are implementation-dependent.”

Clause 9.2.9, page 282, line 2

Replace:

“The `<draw:connector>` element represents a series of lines that are connected to the glue points of two other shapes.”

with

“The `<draw:connector>` element represents a connected set of one or more lines, that visually connects a start and an end point.

Start and/or end points can be assigned to glue points (see section 9.2.19) or absolute positions.”

Clause 9.2.11, page 286, line 13

Insert at end of paragraph:

“The layout of the measure shape is implementation-dependent.”

Clause 9.3, “Relative Sizes”, page 299, line 21

Replace:

“The value `scale-min` equals the value `scale`,”

with

“The value `scale-min` equals the value `scale`,”

Clause 9.4, page 316, line 21

Insert before the sub-clause heading the following new paragraph:

“3D shapes are used to define geometry inside coordinate systems. All ODF coordinate systems are right-handed. Geometry is represented by a tree of scenes. Each scene may contain zero or more 3D shapes, with scene being a special case of shape. The root scene defines the world coordinate system. Each 3D shape may define a local coordinate system relative to its parent 3D scene.”

Clause 9.4.5, page 322, line 32

Replace:

“The `<dr3d:extrude>` element represents a three-dimensional extrude based on a polygon.”

with:

“The `<dr3d:extrude>` element represents a three-dimensional extruded object based on the given polygon. The 3D geometry is formed by extruding the polygon perpendicular to the plane of the polygon by a depth (see `dr3d:depth`, 15.22.6), thus creating a front face, a back face and side faces. The side faces are formed by connecting all corresponding single edges of the front and back plane. When no depth is given, the polygon forms a shape without depth. The closing of the front face depends on `dr3d:close-front` (15.22.9). The closing of the back face depends on `dr3d:close-back` (15.22.10). The size of the back face can be relative to the size of the front face when `dr3d:back-scale` (15.22.5) is used. The side faces defined by the extrusion of the shape are always closed.”

Clause 9.4.6, page 323, line 3

Replace:

“The `<dr3d:rotate>` element represents a three-dimensional rotation shape based on a polygon.”

with:

“The `<dr3d:rotate>` element represents a three-dimensional rotation shape based on the given polygon. The 3D geometry is defined by rotating the XY-Plane with the polygon around the Y-Axis of the local coordinate system. This rotation is linearly interpolated in the given number of steps (see `dr3d:horizontal-segments`) around the given angle (see `dr3d:end-angle`). The side faces are formed by connecting all corresponding single edges of neighboring pairs of rotated polygons. By default, the rotation uses a full rotation (360 degrees, see `dr3d:end-angle` 15.22.8). In that case the 3D geometry is implicitly closed and no front face or back face is created (unless they are of different sizes). If the rotation is different, a front face is created using the first, unrotated polygon and a back face is created using the last rotated polygon. The closing of a front face also depends on `dr3d:close-front` (15.22.9) and back face on `dr3d:close-back` (15.22.10). The size of the back face can be different than the size of the front face when `dr3d:back-scale` (15.22.5) is used.”

Clause 9.5.2, “Extrusion First Light Harsh”, page 328, line 35

Insert before the schema fragment the following new paragraph:

“When the attribute value is `true`, the light is harsh. The light is an undiffused point source. When the attribute value is `false`, the light is not harsh. The means by which the light is softened is implementation-dependent.”

Clause 9.5.2, “Extrusion Second Light Harsh”, page 329, line 1

Insert before the schema fragment the following new paragraph:

“When the attribute value is `true`, the light is harsh. The light is an undiffused point source. When the attribute value is `false`, the light is not harsh. The means by which the light is softened is implementation-dependent.”

Clause 15.4.38, page 578, line 27

Replace:

“Use the `style:text-combine` property to combine characters so that they are displayed within two lines.