INTERNATIONAL STANDARD

ISO/IEC 15475-2

First edition
2002-11-01

Information technology — CDIF transfer format —

Part 2: Syntax SYNTAX.1

Technologies de l'information — Format de transfert CDIF —

Partie 2: Syntaxe SYNTAX.1

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

Reference number
ISO/IEC 15475-2:2002(E)

© ISO/IEC 2002

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

 © ISO/IEC 2002 – All rights reserved

Contents

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

Table of Illustrations

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 15475 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15475-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

ISO/IEC 15475 consists of the following parts, under the general title *Information technology — CDIF transfer format* :

— *Part 1: General rules for syntaxes and encodings*

— *Part 2: Syntax SYNTAX.1*

— *Part 3: Encoding ENCODING.1*

Annex A forms a normative part of this part of ISO/IEC 15475. Annex B is for information only.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

Introduction

This standard will assist the vendors and users of modelling tools and metadata repositories in developing mechanisms for interchanging information. This standard specifies an element of a family of related standards. When used together, these standards specify a mechanism for transferring information between tools.

ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview* and ISO/IEC 15474-2:2002, *Information technology — CDIF framework — Part 2: Modelling and extensibility* should be read first when initially exploring CDIF. The first explains the overall CDIF Architecture and how the family of standards fits together. The second explains the scope, and modelling approach in CDIF. The CDIF Meta-metamodel and extensibility mechanisms are also defined in that document.

ISO/IEC 15475-3:2002, *Information technology — CDIF transfer format — Part 3: Encoding ENCODING.1*, defines an encoding of SYNTAX.1. ISO/IEC 15475-1:2002, *Information technology — CDIF transfer format — Part 1: General rules for syntaxes and encodings* define how CDIF supports multiple exchange syntaxes and encodings.

This document, ISO/IEC 15475-2:2002, *Information technology — CDIF transfer format — Part 2: Syntax SYNTAX.1* defines the CDIF transfer format syntax, SYNTAX.1.

This standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

This document is organized into the following Clauses:

— Clauses 1 to 5 are prescribed ISO/IEC Clauses

— Clause 6: Concepts and facilities

  This specifies the unique identifier for the syntax defined in this document and describes the concept of token separation used in this document.

— Clause 7: Syntax sections and structures in the CDIF transfer

  This section describes, in detail, the exact syntax of each of the three major sections of a transfer.

— Annex A (normative): SYNTAX.1 formal grammar

  This defines the grammar rules for the syntax in Backus Naur Form (BNF).

— Annex B (informative): Multibyte examples

  Supplementary Examples using multibyte codeset encoding are provided.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

Information technology — CDIF transfer format —

Part 2:
Syntax SYNTAX.1

1 Scope

The CDIF family of standards is primarily designed to be used as a description of a mechanism for transferring information between modelling tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF. The language that is defined for the transfer format also has applicability as a general language for Import/Export from repositories. The CDIF semantic metamodel defined for modelling tools also has applicability as the basis of standard definitions for use in repositories.

The standards, which form the complete family of CDIF Standards, are documented in ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview.* These standards cover the overall framework, the transfer format and the CDIF semantic metamodel.

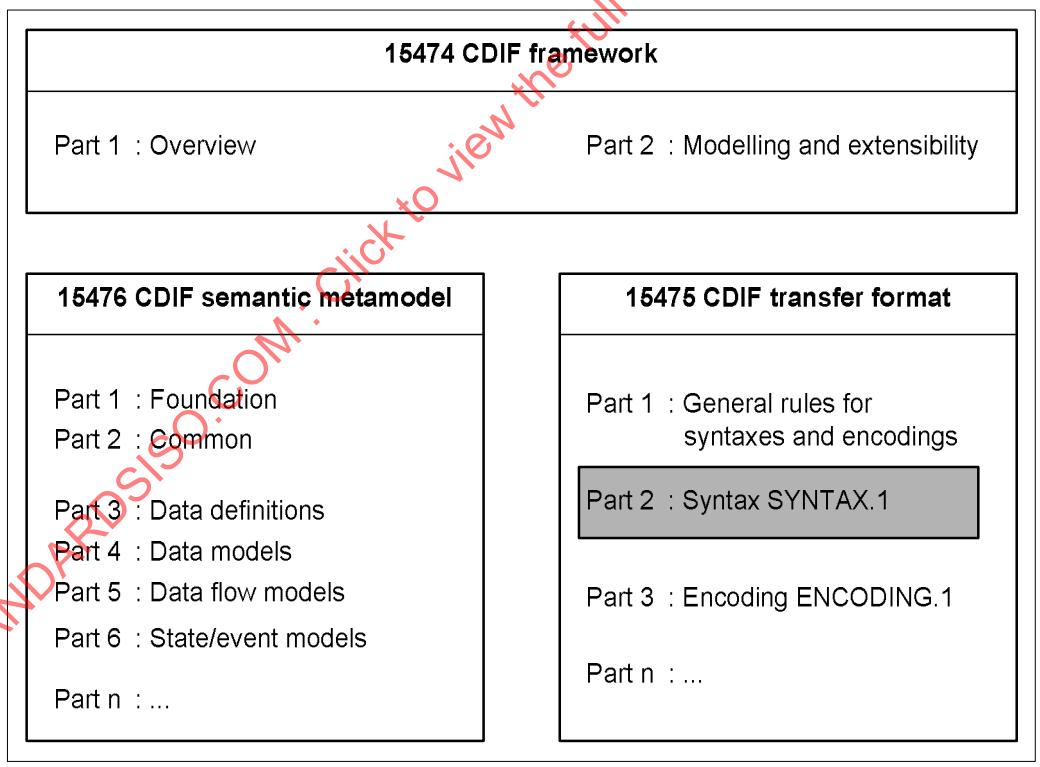

Figure 1 – Position in the CDIF family of standards

The diagram in Figure 1 depicts the various standards that comprise the CDIF family of standards. The shaded box depicts this Standard and its position in the CDIF family of standards.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

This document describes the standard CDIF transfer syntax. No encodings for SYNTAX.1 are specified in this document. ISO/IEC 15475-3:2002, *Information technology — CDIF transfer format — Part 3: Encoding ENCODING.1*, specifies one standard encoding for this syntax.

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides an introduction to the entire CDIF family of standards. It is suitable for:

— Those evaluating CDIF,

— Those who wish to understand the principles and concepts of a CDIF transfer, and

— Those developing importers and exporters.

The documents ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview* and ISO/IEC 15474-2:2002, *Information technology — CDIF framework — Part 2: Modelling and extensibility* should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF family of standards.

This document should be read in conjunction with ISO/IEC 15475-1:2002, *Information technology — CDIF transfer format — Part 1: General rules for syntaxes and encodings*.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

— Entity-Relationship-Attribute modelling;

— Modelling (CASE) tools;

— Information repositories;

— Data dictionaries;

— Multiple meta-layer modelling;

— Formal grammars;

— Transfer formats.

2 Conformance

A product is standards conformant this standard if and only if the product obeys all definitions and rules in Annex A of this standard, and is also CDIF architecture conformant, as defined in Clause 2 of ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview*. A product can be either transfer format standards conformant or non-conformant. Partial standards conformance to a standard defining a part of the CDIF transfer format is not defined.

A product is standards conformant to a CDIF encoding standard only if it is standards conformant to Annex A of ISO/IEC 15475-3:2002, *Information technology — CDIF transfer format — Part 3: Encoding ENCODING.1* and also conformant to Annex A of ISO/IEC 15475-2:2002, *Information technology — CDIF transfer format — Part 2: Syntax SYNTAX.1*. A product is standard conformant to a CDIF syntax standard only if it is standards conformant to Annex A of ISO/IEC 15475-2:2002, *Information technology — CDIF transfer format — Part 2: Syntax SYNTAX.1*.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 15475. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 15475 are encouraged to

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9075:1992, *Information technology — Database languages — SQL*

ISO/IEC 10646-1:1993, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*

ISO/IEC 10646-1:1993/Amd:2:1996, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane. Amendment 2: UCS Transformation Format 8 (UTF-8)*

ISO/IEC 13238-1:—[1)], *Information technology — Data management export/import — Part 1: Standardization framework*

ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview*

ISO/IEC 15474-2:2002, *Information technology — CDIF framework — Part 2: Modelling and extensibility*

ISO/IEC 15475-1:2002, *Information technology — CDIF transfer format — Part 1: General rules for syntaxes and encodings*

4 Terms and definitions

For the purposes of this part of ISO/IEC 15475, the following definitions apply. Unless otherwise noted, the definitions are specific to this part of ISO/IEC 15475.

4.1 From other standards

4.1.1 ISO/IEC 15474-1

This part of ISO/IEC 15475 makes use of the following terms defined in ISO/IEC 15474-1:

CDIF
CDIF family of standards
CDIF graphical notation
CDIF identifier
CDIF semantic metamodel
CDIF meta-metamodel
CDIF transfer
CDIF transfer format
Character set
Encoding
ENCODING.1
Instance
Meta-attribute
Meta-entity
Metamodel
Meta-object
Meta-relationship
Model
Non-terminal symbol
Production rule
Subject area
Syntax
SYNTAX.1
Terminal symbol

1) To be published.

© ISO/IEC 2002 — All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

Transfer
Transfer format

4.1.2 ISO/IEC 15475-1

This part of ISO/IEC 15475 makes use of the following terms defined in ISO/IEC 15475-1:

Metamodel section
Transfer header

4.1.3 ISO/IEC 13238-1

This part of ISO/IEC 15474 makes use of the following terms from ISO/IEC 13238-1:

Transfer file
CDIF transfer file
Export process
Exporter
Import process
Importer
Clear text file encoding

4.2 For this standard

For the purposes of this part of ISO/IEC 15475 new terms are defined when introduced. Double quotes are used to introduce new terms (e.g., “model layer”).

5 Symbols (and abbreviated terms)

5.1 Naming and diagramming notations

All meta-objects and meta-meta-objects in CDIF (in metamodels and meta-metamodels) are named by concatenating all the words that name the meta-object or meta-meta-object; the first letter of each word is upper-case, the rest are lower-case (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*).

Full details of the CDIF graphical notation used in the metamodel and the meta-metamodel can be found in the Framework document (ISO/IEC 15474-2:2002).

5.2 BNF conventions

An extended Backus Naur Form (BNF) is used to define the structure and describe the sequence of data in the transfer file.

This document uses the conventions established in subclause 5.2 of ISO/IEC 15475-1:2002, *Information technology — CDIF transfer format — Part 1: General rules for syntaxes and encodings*.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

5.3 Abbreviations

The following abbreviations are used in this part of ISO/IEC 15475:

BNF Backus Naur Form

CDIF CASE Data Interchange Format (originally)

6 CDIF transfer concepts and facilities

6.1 Syntax identifier

The syntax defined in this version of this standard shall have a CDIF standardized syntax identifier of "SYNTAX.1", and a version of "15475-2:2002". This is used to define the standardized syntax used in a CDIF transfer (see ISO/IEC 15475-1).

The syntax of the CDIF syntax ID and the syntax Version is as follows:

```
<SyntaxID> ::=          'SYNTAX.1'
<SyntaxVersion> ::=     '15475-2:2002'
```

6.2 Token separation rules

The "tokens" (or terminal symbols) are the primitive level elements in the syntax. Tokens are normally separated by one or more "whitespace" characters, as defined in the specific encoding. After token identification, these whitespace characters are ignored by importers.

Certain terminal symbols act as token separators; these terminal symbols do not need to be preceded or followed by whitespace characters. All other terminal symbols must be followed by one of these token separators or by one or more whitespace characters. The terminal symbols that act as token separators are listed in subclause 7.7 Syntax terminal symbols.

7 Syntax sections and structures in the CDIF transfer

7.1 Introduction

This section describes, in detail, the exact syntax of each of the major sections of a transfer, together with the syntax of each of the component structures.

Examples of transfers given in this section use the CDIF standard encoding as defined in ISO/IEC 15475-3.

7.2 CDIF transfer components

7.2.1 Introduction

The general syntax of a CDIF transfer is as follows:

```
<CDIFTransfer> ::=      <TransferHeader>
                        <TransferContents>
```

Figure 2 illustrates the complete structure of a CDIF transfer.

A transfer is typically contained in a file, but other means of transfer such as pipes, mailboxes, shared memory, or any other mechanism that can be interpreted as a byte stream may be used.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

7.2.2 Transfer header

The transfer header consists of the CDIF Signature, the syntax identifier and the encoding identifier. As the transfer header syntax is the same for any CDIF transfer, it is defined in ISO/IEC 15475-1. The syntax identifier for SYNTAX.1 is specified in 6.1 Syntax identifier of this document.

7.2.3 Transfer contents

As the first level of the grammar of the transfer Contents is the same for any CDIF transfer, it is defined in ISO/IEC 15475-1, but is reproduced here for clarity:

```
<TransferContents> ::=    <HeaderSectionClause>
                          <MetaModelSectionClause>
                          [ <ModelSectionClause> ]
```

The further levels are detailed in the next section.

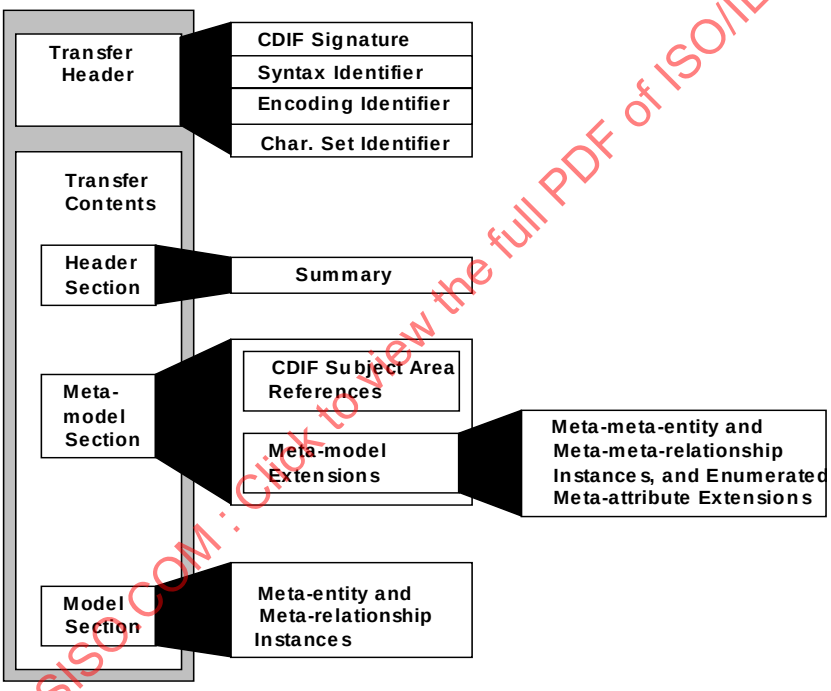

Figure 2 – CDIF transfer

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

7.3 Header section

7.3.1 Introduction

The header section defines information that applies to the whole transfer.

The syntax of the header section is:

```
<HeaderSectionClause> ::= <OpenScope>
                              <HeaderKeyword>
```


© ISO/IEC 2002 — All rights reserved

```
                    <SummaryClause>
                <CloseScope>
```

that is, an introductory keyword followed by a summary , all enclosed within scope delimiters.

An example of the header section is:

```
(:HEADER <SummaryClause>)
```

7.3.2 Summary

The summary specifies summary information about the transfer, in the form of a number of items. Each item is introduced by an identifier. This standard defines meanings for certain summary identifiers. Other identifiers may be used, to introduce other items of summary information, but their meanings are not defined in this standard.

The syntax of the summary is:

```
<SummaryClause> ::=       <OpenScope>
                              <SummaryKeyword>
                              [ <IdentifierValuePair> ] ...
                          <CloseScope>
<IdentifierValuePair> ::= <OpenScope>
                              <SummaryIdentifier> <StringValue>
                          <CloseScope>
<SummaryIdentifier> ::=   <Identifier>
<StringValue> ::=         <String>
```

The identifiers must conform to the rules for data type identifier.

The defined summary identifiers and their definitions are:

- ExporterName - The name of the exporting software product;
- ExporterVersion - The version of the exporting software product;
- ExportDate - The date of commencement of the export, in the form YYYY/MM/DD;
- ExportTime - The time of commencement of the export, in the form HH:MM:SS;
- PublisherName - The name of the person (or function) who initiated the export.

An example of a summary is:

```
(:SUMMARY                 (ExporterName       "AutoPal")
                          (ExporterVersion    "3.2")
                          (ExportDate         "1991/01/12")
                          (ExportTime         "13:00:00")
                          (PublisherName      "Andrew Shilling")
                          (HardwarePlatform   "Apple MacIntosh")
                          (VendorCompanyName  "UltraCASE")
                      )
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

7.4 Metamodel section

7.4.1 Introduction

The metamodel section of the transfer consists of references to standardized subject areas, followed by extensions to the metamodel. The metamodel for the transfer, known as the "working metamodel", is defined by the set of meta-meta-entity and meta-meta-relationship instances that are used in any of the referenced subject areas, plus those added by extensions.

The working metamodel may be extended as defined in ISO/IEC 15474-2.

As a conveyor of metamodels, this section contains instances of meta-metamodel concepts. The mapping between these concepts, as defined in ISO/IEC 15474-2, and this syntax is straightforward and therefore clarification is included only where necessary.

The syntax of the metamodel section is:

```
<MetaModelSectionClause> ::=
                          <OpenScope>
                               <MetaModelKeyword>
                               <CDIFSubjectAreaReferenceClause>...
                               [ <MetaModelExtensionClause> ]...
                          <CloseScope>
```

An example of the metamodel section is:

```
     (:META-MODEL         <CDIFSubjectAreaReferenceClause>...
                          [ <MetaModelExtensionClause> ]...
                          )
```

7.4.2 CDIF subject area reference

The syntax of the CDIF Subject Area Reference is:

```
<CDIFSubjectAreaReferenceClause> ::=
                          <OpenScope>
                               <SubjectAreaReferenceKeyword>
                               <SubjectAreaName>
                               <OpenScope>
                                    <VersionNumberKeyword>
                                    <SubjectAreaVersionNumber>
                               <CloseScope>
                          <CloseScope>
<SubjectAreaName> ::=     <MetaObjectName>
                          !!<SubjectAreaName> must be the value of the Name meta-meta-
                          attribute of the relevant SubjectArea instance.
<SubjectAreaVersionNumber> ::=
                          <String>
                          !! <SubjectAreaVersionNumber> must be the value of the
                          VersionNumber meta-meta-attribute of the relevant SubjectArea
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

```
                    instance. A reference to at least one CDIF Subject Area must
                    be included in the metamodel section.
```

An example of the CDIF Subject Area Reference is:

```
(:SUBJECTAREAREFERENCE Foundation
                    (:VERSIONNUMBER "15476-1:2002")
)
```

7.4.3 Metamodel extension

The syntax of the metamodel Extension is:

```
<MetaModelExtensionClause> ::=
                     <MetaMetaEntityInstance>
                   | <MetaMetaRelationshipInstance>
                   | <EnumeratedMetaAttributeExtension>
```

Note that although the syntax allows meta-meta-entity instances, meta-meta-relationship instances and enumerated meta-attribute extensions to be placed in arbitrary order, no forward references are allowed in a transfer.

7.4.4 Meta-meta-entity instance

Instances of meta-meta-entities are specified by the name of the meta-meta-entity (e.g., *MetaEntity*, *MetaRelationship*, *MetaAttribute* or *SubjectArea*), followed by a unique CDIF meta-identifier for the instance, followed by the names and values of the other meta-meta-attributes for the meta-meta-entity.

The syntax of the meta-meta-entity instance is:

```
<MetaMetaEntityInstance> ::=
                    <OpenScope>
                        <MetaMetaEntityName> <CDIFMetaIdentifier>
                        [ <MetaMetaAttributeInstance> ] ...
                    <CloseScope>
<MetaMetaEntityName> ::=
                    <MetaMetaObjectName>
                    !! <MetaMetaEntityName> must be the name of a meta-meta-
                    entity, as defined in ISO/IEC 15474-2, e.g., MetaEntity,
                    MetaRelationship, MetaAttribute or SubjectArea.
<CDIFMetaIdentifier> ::=
                    <Identifier>
                    !! CDIF meta-identifiers beginning with a numeric character
                    are reserved for use in the CDIF semantic metamodel.
```

A meta-meta-attribute instance is used to represent each of the meta-meta-attributes (other than the *CDIFMetaIdentifier* meta-meta-attribute) of the meta-meta-entity. <MetaMetaAttributeInstance> is defined in the next section.

Several examples of the meta-meta-entity instance are:

```
(1)   (SubjectArea NEW00001     [ <MetaMetaAttributeInstance> ] ... )
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

```
(2)   (MetaEntity ME001
                                [ <MetaMetaAttributeInstance> ] ...
                                )
(3)   (MetaAttribute MA001
                                [ <MetaMetaAttributeInstance> ] ...
                                )
(4)   (MetaRelationship MR001
                                [ <MetaMetaAttributeInstance> ] ...
                                )
```

7.4.5 Meta-meta-attribute instance

Instances of meta-meta-attribute (other than the *CDIFMetaIdentifier* meta-meta-attribute) are specified by a sequence consisting of the name of the meta-meta-attribute followed by its value.

The syntax of the meta-meta-attribute instance is:

```
<MetaMetaAttributeInstance> ::=
                            <OpenScope>
                                 <MetaMetaAttributeName> <MetaMetaAttributeValue>
                            <CloseScope>
<MetaMetaAttributeName> ::=
                            <MetaMetaObjectName>
<MetaMetaAttributeValue> ::=
                            <MetaAttributeValue>
                            !! <MetaMetaAttributeName> must be the name of a meta-meta-
                            attribute associated with the meta-meta-entity, as defined in
                            ISO/IEC 15474-2.
```

<MetaAttributeValue> is defined in 7.5.5 Meta-attribute value.

Several examples of the meta-meta-attribute instance are:

```
(1)   (Name *Function*)
(2)   (Description #[This defines a Business Function which is associated with a
                            particular Process or Processes in a Data Flow Model]#)
(3)   (Aliases "Business Function,Activity")
(4)   (IsOptional -True-)
```

7.4.6 Meta-meta-relationship instance

Instances of meta-meta-relationships are specified by a sequence consisting of the name of the meta-meta-relationship and the CDIF meta-identifiers for the source and destination meta-meta-entities. *CollectableMetaObject.IsUsedIn.SubjectArea*, *MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject*, *AttributableMetaObject.HasSubtype.AttributableMetaObject*, *MetaRelationship.HasSource.MetaEntity*, and *MetaRelationship.HasDestination.MetaEntity* are full names of meta-meta-relationships.

There shall be no forward references to meta-meta-entities defined later in the metamodel section.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

The syntax of the meta-meta-relationship instance is:

```
<MetaMetaRelationshipInstance> ::=
                              <OpenScope>
                                   <FullMetaMetaRelationshipName>
                                   <SourceMetaMetaEntityCDIFMetaIdentifier>
                                   <DestinationMetaMetaEntityCDIFMetaIdentifier>
                              <CloseScope>
<FullMetaMetaRelationshipName> ::=
                              <SourceMetaMetaEntityName> <Dot>
                              <MetaMetaRelationshipName> <Dot>
                              <DestinationMetaMetaEntityName>
                              !! <FullMetaMetaRelationshipName> must be the full name of a
                              meta-meta-relationship, as defined in ISO/IEC 15474-2.
<SourceMetaMetaEntityName> ::=
                              <MetaMetaObjectName>
<MetaMetaRelationshipName> ::=
                              <MetaMetaObjectName>
<DestinationMetaMetaEntityName> ::=
                              <MetaMetaObjectName>
<SourceMetaMetaEntityCDIFMetaIdentifier> ::=
                              <CDIFMetaIdentifier>
<DestinationMetaMetaEntityCDIFMetaIdentifier> ::=
                              <CDIFMetaIdentifier>
<CDIFMetaIdentifier> ::=  <Identifier>
```

Several examples of the meta-meta-relationship instance are:

```
(1)   (MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
                              MYID002 XYZ001)
!! This adds a meta-meta-relationship instance to make the MetaAttribute with
                              CDIFMetaIdentifier MYID002 a local meta-attribute of the
                              MetaEntity or MetaRelationship with CDIFMetaIdentifier XYZ001.
(2)   (CollectableMetaObject.IsUsedIn.SubjectArea
                              MATT006 SUBJ001)
```

7.4.7 Enumerated meta-attribute extension

The syntax of the enumerated meta-attribute extension is:

```
<EnumeratedMetaAttributeExtension> ::=
                              <OpenScope>
                                   <ExtendMetaAttributeKeyword> <CDIFMetaIdentifier>
                                   <OpenScope>
                                        <EnumeratedValue>
                                        [<EnumeratedSeparator>
                                        <EnumeratedValue>]...
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

```
                                <CloseScope>
                          <CloseScope>
<CDIFMetaIdentifier> ::=  <Identifier>
                          !! <CDIFMetaIdentifier> must identify a MetaAttribute with
                          Enumerated data type defined in the standardized model or
                          earlier in the metamodel Extension Clause. The Enumerated
                          Values are appended to those values that have already been
                          defined for the MetaAttribute.
```

An example of the enumerated meta-attribute extension is:

```
(:EXTENDMETA-ATTRIBUTE MA001 (<Encrypted>,<Nonencrypted>))
```

7.5 Model section

7.5.1 Introduction

The model section contains references to attributable meta-objects (object types) and actual model data. The object types referenced here are instances of *MetaEntity* and *MetaRelationship* that were defined in the metamodel section as part of the CDIF Subject Area references or in the metamodel extension. The model data are in the form of meta-entity instance, meta-relationship instance and meta-attribute instance.

As a conveyor of models, this section contains instances of metamodel concepts. The mapping between these concepts, as defined in the CDIF semantic metamodel, and this syntax is straightforward and therefore clarification is included only where necessary.

All meta-entity and meta-relationship instances in the Model section are uniquely identified by the *CDIFIdentifier* meta-attribute. The exporter is responsible for ensuring the uniqueness of these identifiers. An importer does not have to retain them, having completed the importation process (successfully or unsuccessfully).

The syntax of the Model section is:

```
<ModelSectionClause> ::=  <OpenScope>
                               <ModelKeyword> <ObjectClause>...
                          <CloseScope>
<ObjectClause> ::=
                          <MetaEntityInstance> | <MetaRelationshipInstance>
```

An example of the model section is:

```
(:MODEL <ObjectClause> ... )
```

7.5.2 Meta-entity instance

Instances of meta-entity are specified by a sequence consisting of the name of the meta-entity (as defined in the metamodel) and a unique identifier for the meta-entity instance, followed by the names and values of any meta-attributes of the meta-entity (as defined in the metamodel).

The syntax of the meta-entity instance is:

```
<MetaEntityInstance> ::=  <OpenScope>
                               <MetaEntityName> <CDIFIdentifier>
                               [ <MetaAttributeInstance> ] ...
                          <CloseScope>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

```
<MetaEntityName> ::=
                              <MetaObjectName>
<CDIFIdentifier> ::=
                              <Identifier>
```

An example of the meta-entity instance is:

```
(DataModel MOD01     [ <MetaAttributeInstance> ...` )
```

7.5.3 Meta-relationship instance

Instances of meta-relationship are specified by a sequence consisting of the name of the MetaRelationship (as defined in the metamodel), a unique identifier for the instance, a source meta-entity identifier, a destination meta-entity identifier, and the names and values of any meta-attributes of the meta-relationship (as defined in the metamodel).

There shall be no forward references to meta-entities defined later in the model section.

The syntax of the meta-relationship instance is:

```
<MetaRelationshipInstance> ::=
                              <OpenScope>
                                   <FullMetaRelationshipName>
                                   <MetaRelationshipCDIFIdentifier>
                                   <SourceMetaEntityCDIFIdentifier>
                                   <DestinationMetaEntityCDIFIdentifier>
                                   [ <MetaAttributeInstance> ] ...
                              <CloseScope>
<FullMetaRelationshipName> ::=
                              <SourceMetaEntityName> <Dot>
                              <MetaRelationshipName> <Dot>
                              <DestinationMetaEntityName>
<SourceMetaEntityName> ::=<MetaObjectName>
<MetaRelationshipName> ::=<MetaObjectName>
<DestinationMetaEntityName> ::=
                              <MetaObjectName>
!!<SourceMetaEntityName>, <MetaRelationshipName> and <DestinationMetaEntityName> must be
                              the names used in the definition of the meta-relationship.
<MetaRelationshipCDIFIdentifier> ::=
                              <CDIFIdentifier>
<SourceMetaEntityCDIFIdentifier> ::=
                              <CDIFIdentifier>
<DestinationMetaEntityCDIFIdentifier> ::=
                              <CDIFIdentifier>
<CDIFIdentifier> ::=
                              <Identifier>
```

An example of the meta-relationship instance is:

```
(DataModel.IsCollectionOf.DataModelObject R001 MOD01 ENT02)
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

7.5.4 Meta-attribute instance

Instances of meta-attribute (other than the *CDIFIdentifier* meta-attribute) are specified by a sequence consisting of the name of the meta-attribute (as defined in the metamodel), followed by its value.

The syntax of the meta-attribute instance is:

```
<MetaAttributeInstance> ::<OpenScope>
                              <MetaAttributeName> <MetaAttributeValue>
                          <CloseScope>
<MetaAttributeName> ::=   <MetaObjectName>
```

An example of the meta-attribute instance is:

```
    (Name <MetaAttributeValue>)
```

7.5.5 Meta-attribute value

7.5.5.1 Introduction

The syntax of the meta-attribute value is:

```
<MetaAttributeValue> ::=  <BitmapValue> | <BooleanValue> | <DateValue>
                          | <EnumeratedValue> | <FloatValue> | <IdentifierValue>
                          | <IntegerValue> | <IntegerListValue> | <PointValue>
                          | <PointListValue> | <StringValue> | <TextValue>
                          | <TimeValue>
```

7.5.5.2 Bitmap value

A bitmap value is defined as a list of pixel values, preceded by the height and width dimensions of the bitmap. A pixel value is a red/green/blue triple representing the intensity of the respective colours. Pixel values in the bitmap are ordered left-to-right and top-to-bottom.

The syntax of the bitmap value is:

```
<BitmapValue> ::=         <BitmapKeyword> <Height> <Width>
                          <OpenScope>
                              <Bitmap>
                          <CloseScope>
<Height> ::=              <HeightKeyword> <PositiveInteger>
<Width> ::=               <WidthKeyword> <PositiveInteger>
<Bitmap> ::=              <PixelValue> [ <ListSeparator> <PixelValue> ] ...
<PixelValue> ::=          <OpenScope>
                              <PixelRedIntensity> <PixelSeparator>
                              <PixelGreenIntensity> <PixelSeparator>
                              <PixelBlueIntensity>
                          <CloseScope>
<PixelRedIntensity> ::=   <PixelIntensity>
<PixelGreenIntensity> ::= <PixelIntensity>
<PixelBlueIntensity> ::=  <PixelIntensity>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

The following are examples of valid and invalid bitmap values:

```
(1)   :BITMAP :HEIGHT 2 :WIDTH 2
                        ((120,50,35),(130,80,70),
                        (100,28,231),(111,255,0)) - Valid
(2)   :BITMAP :HEIGHT 1 :WIDTH 2
                        ((120,50,35),(130,80,70)) - Valid
(3)   :BITMAP :HEIGHT 2 :WIDTH 2
                        ((255,255,255),(255,255,255),
                        (255,255,255),(255,255,255))    - Valid
(4)   :BITMAP ((120,50,35),(130,80,70),
                        (100,28,231),(111,255,0)) - Invalid: missing height and width
(5)   :HEIGHT 1 :WIDTH 2
                        ((120,50,35),(130,80,70)) - Invalid: missing bitmap keyword
```

7.5.5.3 Boolean value

The syntax of the boolean value is:

```
<BooleanValue> ::=        <TrueValue> | <FalseValue>
```

The following are examples of the boolean value:

```
(1)   -TRUE-
(2)   -FALSE-
```

7.5.5.4 Date value

The syntax of the date value is:

```
<DateValue> ::=           <DateKeyword> <Date> <DateClassValue>
<DateClassValue> ::=      <Absolute> | <RelativePositive> | <RelativeNegative>
```

The following are examples of valid and invalid date values:

```
(1)   :DATE 1940/12/07 Absolute           - Valid
(2)   :DATE 1920/07/20                    - Invalid: <DateClassValue> missing
(3)   :DATE 0002/10/11 RelativePositive   - Valid
(4)   1940/12/07 Absolute                 - Invalid: date keyword missing
```

7.5.5.5 Enumerated value

```
<EnumeratedValue> ::=     !! It is a terminal symbol in the syntax. Its representation
                          is specified in the encoding.
```

An example of the enumerated value is:

```
        <Red>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

7.5.5.6 Float value

A float value is a decimal number with up to 16 decimal digits of precision within the range of 10^{-1023} to 10^{1023}.

```
<FloatValue> ::=     !! It is a terminal symbol in the syntax. Its representation is
                         specified in the encoding.
```

Several examples of valid float values are:

```
(1)  #f4E9
(2)  #f123.45E2
(3)  #f-123.45E15
(4)  #f+0.23E5
(5)  #f0.23E-3
(6)  #f0.23E+3
```

7.5.5.7 Identifier value

```
<IdentifierValue> ::=     !! It is a terminal symbol in the syntax. Its representation
                          is specified in the encoding.
```

An example of the identifier value is:

```
*johnBrownsBody*
```

7.5.5.8 Integer value

Integer values are expressed as decimal values, binary values, hexadecimal values or octal values.

The syntax of the integer value is:

```
<IntegerValue> ::=        <DecimalIntegerValue> | <BinaryValue> | <HexaDecimalValue>
                          | <OctalValue>
```

The following are examples of the integer values:

```
(1)  #d12345          -    positive decimal
(2)  #d-12345         -    negative decimal
(3)  #d+12345         -    positive decimal
(4)  #b10101          -    positive binary
(5)  #b-10101         -    negative binary
(6)  #o31727          -    positive octal
(7)  #o-31727         -    negative octal
(8)  #h1e2cf          -    positive hexadecimal
(9)  #hffffffff       -    positive hexadecimal
(10) #h-ffffffff      -    negative hexadecimal
```

7.5.5.9 Integer list value

An integer list is defined as a sequence of integers.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

The syntax of the integer list value is:

```
<IntegerListValue> ::=     <IntegerListKeyword>
                           <OpenScope>
                                <IntegerValue> [ <ListSeparator> <IntegerValue> ] ...
                           <CloseScope>
```

The following are valid and invalid examples of the integer list value:

```
(1)  :INTEGERLIST (#d10,#d20,#d11,#d15,#d26,#d32)   - Valid
(2)  :INTEGERLIST (#b10101,#h32fe,#o37712)          - Valid
(3)  :INTEGERLIST (#d10,#d2.4) - Invalid: 2.4 is not an integer value
(4)  (#d10,#d20,#d11,#d15, #d32)     - Invalid: missing integer list keyword
```

7.5.5.10 Point value

A point value is defined by three integer values representing x, y and z coordinates.

The syntax of the point value is:

```
<PointValue> ::=
                           <PointKeyword> <Point>
<Point> ::=                <OpenScope>
                                <XValue> <PointSeparator>
                                <YValue> <PointSeparator>
                                <ZValue>
                           <CloseScope>
<XValue>   ::=             <Integer>
<YValue>   ::=             <Integer>
<ZValue>   ::=             <Integer>
```

The following are valid and invalid examples of the point value:

```
(1)  :POINT(0 0 0)          -    Valid
(2)  :POINT(1 303 292)      -    Valid
(3)  :POINT(20 30)          -    Invalid: three coordinates must be supplied
(4)  :POINT(qq eo 50)       -    Invalid: must be integer values
```

7.5.5.11 Point list value

A point list is defined as a sequence of points.

The syntax of the point list value is:

```
<PointListValue> ::=       <PointListKeyword>
                           <OpenScope>
                                <Point> [ <ListSeparator> <Point> ] ...
                           <CloseScope>
<Point> ::=                <OpenScope>
                                <XValue> <PointSeparator>
                                <YValue> <PointSeparator>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

```
                                    <ZValue>
                                <CloseScope>
```

The following are examples of the point list value:

```
(1)   :POINTLIST ((0 0 0),(1 1 1),(3 3 3))
                                - Valid
(2)   :POINTLIST ((1 303 292),(321 22 33))
                                - Valid
(3)   :POINTLIST 12)            - Invalid: does not have an <OpenScope>
(4)   ((0 0 0),(1 1 1))         - Invalid: point list keyword missing
```

7.5.5.12 String value

A string value is defined as a character string.

The syntax of the string value is:

```
<StringValue> ::=           <String>
```

An example of the string value is:

```
"This is a string"
```

7.5.5.13 Text value

The text data type is represented as a group of characters, not necessarily printable nor of any pre-determined maximum length. It is specified as a sequence of one or more text strings; the text value itself is the concatenation of the contents of the individual text strings.

The syntax of the text value is:

```
<TextValue> ::=             <TextString> [ <ListSeparator> <TextString> ] ...
```

The specific encoding being used may constrain the maximum length of an individual text string.

An example of the text value is[2]:

```
(1)   #[Program SumIntegers(Input,Output);
      var  total,input_integer : Integer;
      begin
           while not EOF(Input) do
                begin
                ReadLn(input_integer);
                total:=total+input_integer
                end;
           WriteLn('Total =',total);
                           end.]#
                           -    Valid
```

[2] This example of the text value subclause has been laid out with whitespace characters, such as <CR>, transformed for presentation.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

7.5.5.14 Time value

The syntax of the time value is:

```
<TimeValue> ::=      <TimeKeyword> <Time> <TimeClassValue>
<TimeClassValue> ::=      <AbsoluteUTC> | <AbsoluteLocal> | <RelativePositive> |
                          <RelativeNegative>
```

The following are examples of valid and invalid time values:

```
(1)   :TIME 07:20:23 AbsoluteUTC
                          - Valid
(2)   :TIME 29:25:19 AbsoluteLocal
                          - Invalid: illegal hour value 29
(3)   :TIME 03:59:59
                          - Invalid: <TimeClassValue> missing
(4)   :TIME 00:00:00.250 RelativePositive
                          - Valid
(5)   07:20:23 AbsoluteUTC
                          - Invalid: time keyword missing
```

7.6 Comment

Comments may appear anywhere in the syntax between any two terminal symbols. Comments consist of a sequence of printable characters. Embedded whitespace characters are allowed.

```
<Comment>::=         !! It is a terminal symbol in the syntax. Its representation is
                          specified in the encoding.
```

The specific encoding being used may constrain the maximum length of a comment.

The following are examples of valid and invalid comments[3]:

```
(1)   #| this is
                            a multi-line
                            comment
                          |#                        - Valid
(2)   ( 1 2 #| buckle
                              my shoe |# 3 4) - Valid (in line comment)
(3)   #| hello ... <eof>        - Invalid (unbalanced delimiters)
```

7.7 Syntax terminal symbols

The following are the terminal symbols in this syntax. Their representation is dependent on the specific encoding used. ENCODING.1, ISO/IEC 15475-3, has been used for the examples in this document.

```
<Absolute>
```

---

3) These examples of the comment subclause have been laid out with whitespace characters, such as <CR>, transformed for presentation. <eof> is used to represent the end of the file.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

<AbsoluteUTC>
<AbsoluteLocal>
<BinaryValue>
<BitmapKeyword>
<CloseScope>
<Comment>
<Date>
<DateKeyword>
<DecimalIntegerValue>
<Dot>
<EnumeratedSeparator>
<EnumeratedValue>
<ExtendMetaAttributeKeyword>
<FalseValue>
<FloatValue>
<HeaderKeyword>
<HeightKeyword>
<HexaDecimalValue>
<Identifier>
<IdentifierValue>
<Integer>
<IntegerListKeyword>
<ListSeparator>
<MetaModelKeyword>
<MetaMetaObjectName>
<MetaObjectName>
<ModelKeyword>
<OctalValue>
<OpenScope>
<PixelIntensity>
<PixelSeparator>
<PointKeyword>
<PointListKeyword>
<PointSeparator>
<PositiveInteger>
<RelativeNegative>
<RelativePositive>
<String>
<SubjectAreaReferenceKeyword>
<SummaryKeyword>
<TextString>
<Time>
<TimeKeyword>
<TrueValue>

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

<VersionNumberKeyword>
<WidthKeyword>

The following terminal symbols are used as keywords in the grammar of this syntax:

<BitmapKeyword>
<DateKeyword>
<ExtendMetaAttributeKeyword>
<HeaderKeyword>
<HeightKeyword>
<IntegerListKeyword>
<MetaModelKeyword>
<ModelKeyword>
<PointKeyword>
<PointListKeyword>
<SubjectAreaReferenceKeyword>
<SummaryKeyword>
<TimeKeyword>
<VersionNumberKeyword>
<WidthKeyword>

The following terminal symbols act as token separators in this syntax:

<CloseScope>
<Dot>
<EnumeratedSeparator>
<ListSeparator>
<OpenScope>
<PixelSeparator>
<PointSeparator>

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved

Annex A
(normative)

SYNTAX.1 formal grammar

This section contains a collection of the syntax presented in 7 syntax structures in the CDIF transfer. This BNF alone is not a complete definition of SYNTAX.1. There are additional definitions and restrictions expressed in 6 CDIF transfer concepts and facilities and in the text in 7 Syntax sections and structures in the CDIF transfer.

```
<TransferContents> ::=      <HeaderSectionClause>
                            <MetaModelSectionClause>
                            [ <ModelSectionClause> ]
<HeaderSectionClause> ::= <OpenScope>
                                  <HeaderKeyword>
                                  <SummaryClause>
                            <CloseScope>
<SummaryClause> ::=         <OpenScope>
                                  <SummaryKeyword>
                                  [ <IdentifierValuePair> ] ...
                            <CloseScope>
<IdentifierValuePair> ::= <OpenScope>
                                  <SummaryIdentifier> <StringValue>
                            <CloseScope>
<SummaryIdentifier> ::=     <Identifier>
<StringValue> ::=           <String>
<MetaModelSectionClause> ::=
                            <OpenScope>
                                  <MetaModelKeyword>
                                  <CDIFSubjectAreaReferenceClause>...
                                  [ <MetaModelExtensionClause> ]...
                            <CloseScope>
<CDIFSubjectAreaReferenceClause> ::=
                            <OpenScope>
                                  <SubjectAreaReferenceKeyword>
                                  <SubjectAreaName>
                                  <OpenScope>
                                        <VersionNumberKeyword>
                                        <SubjectAreaVersionNumber>
                                  <CloseScope>
                            <CloseScope>
<SubjectAreaName> ::=       <MetaObjectName>
                            !!<SubjectAreaName> must be the value of the Name meta-meta-
                            attribute of the relevant SubjectArea instance.
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15475-2:2002

© ISO/IEC 2002 — All rights reserved