INTERNATIONAL STANDARD

ISO/IEC 29500-1

Second edition
2011-08-15

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

Part 1: Fundamentals and Markup Language Reference

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

Partie 1: Principes essentiels et référence de langage de balisage

Reference number
ISO/IEC 29500-1:2011(E)

Documents
Document Sharing Hub
© ISO/IEC 2011

Full document details: ISO_IEC 29500-1_2011



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

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

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: ISO_IEC 29500-1_2011

Table of Contents

Foreword ..... **viii**

Introduction ..... **ix**

1. Scope **1**

2. Conformance **2**

2.1 Goal 2
2.2 Issues 2
2.3 What ISO/IEC 29500 Specifies 3
2.4 Document Conformance 3
2.5 Application Conformance 4
2.6 Application Descriptions 5
2.7 Interoperability Guidelines 6

3. Normative References **8**

4. Terms and Definitions **12**

5. Notational Conventions **15**

6. Acronyms and Abbreviations **16**

7. General Description **17**

8. Overview **18**

8.1 Packages and Parts 18
8.2 Consumers and Producers 18
8.3 WordprocessingML 18
8.4 SpreadsheetML 19
8.5 PresentationML 20
8.6 Supporting MLs 21

9. Packages **23**

9.1 Office Open XML's Use of OPC 23
9.2 Relationships in Office Open XML 24

10. Markup Compatibility and Extensibility **29**

10.1 Constraints on Office Open XML's Use of Markup Compatibility and Extensibility 29

11. WordprocessingML **30**

11.1 Glossary of WordprocessingML-Specific Terms 30
11.2 Package Structure 30
11.3 Part Summary 32
11.4 Document Template 59
11.5 Framesets 59
11.6 Master Documents and Subdocuments 60
11.7 Mail Merge Data Source 62
11.8 Mail Merge Header Data Source 62
11.9 XSL Transformation 63

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

12. SpreadsheetML 65
12.1 Glossary of SpreadsheetML-Specific Terms 65
12.2 Package Structure 66
12.3 Part Summary 68
12.4 External Workbooks 102

13. PresentationML 104
13.1 Glossary of PresentationML-Specific Terms 104
13.2 Package Structure 104
13.3 Part Summary 107
13.4 HTML Publish Location 125
13.5 Slide Synchronization Server Location 126

14. DrawingML 128
14.1 Glossary of DrawingML-Specific Terms 128
14.2 Part Summary 128

15. Shared 141
15.1 Glossary of Shared Terms 141
15.2 Part Summary 142
15.3 Hyperlinks 165

16. Part Overview 167
16.1 WordprocessingML Summary 167
16.2 SpreadsheetML Summary 167
16.3 PresentationML Summary 168
16.4 DrawingML Summary 169
16.5 Shared Summary 169

17. WordprocessingML Reference Material 171
17.1 Table of Contents 171
17.2 Main Document Story 191
17.3 Paragraphs and Rich Formatting 199
17.4 Tables 398
17.5 Custom Markup 528
17.6 Sections 608
17.7 Styles 677
17.8 Fonts 748
17.9 Numbering 771
17.10 Headers and Footers 820
17.11 Footnotes and Endnotes 835
17.12 Glossary Document 875
17.13 Annotations 896
17.14 Mail Merge 1043
17.15 Settings 1089
17.16 Fields and Hyperlinks 1298
17.17 Miscellaneous Topics 1442
17.18 Simple Types 1453

18. SpreadsheetML Reference Material 1685

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

18.1 Table of Contents 1685
18.2 Workbook 1704
18.3 Worksheets 1757
18.4 Shared String Table 1902
18.5 Tables 1914
18.6 Calculation Chain 1932
18.7 Comments 1935
18.8 Styles 1943
18.9 Metadata 2001
18.10 Pivot Tables 2018
18.11 Shared Workbook Data 2183
18.12 QueryTable Data 2217
18.13 External Data Connections 2225
18.14 Supplementary Workbook Data 2249
18.15 Volatile Dependencies 2263
18.16 Custom XML Mappings 2269
18.17 Formulas 2277
18.18 Simple Types 2671

19. PresentationML Reference Material 2761

19.1 Table of Contents 2761
19.2 Presentation 2767
19.3 Slides 2814
19.4 Comments 2870
19.5 Animation 2875
19.6 Slide Synchronization Data 2981
19.7 Simple Types 2982

20. DrawingML - Framework Reference Material 3015

20.1 DrawingML - Main 3015
20.2 DrawingML - Picture 3457
20.3 DrawingML - Locked Canvas 3466
20.4 DrawingML - WordprocessingML Drawing 3467
20.5 DrawingML - SpreadsheetML Drawing 3516

21. DrawingML - Components Reference Material 3554

21.1 DrawingML - Main 3554
21.2 DrawingML - Charts 3760
21.3 DrawingML - Chart Drawings 3923
21.4 DrawingML - Diagrams 3951

22. Shared MLs Reference Material 4084

22.1 Math 4084
22.2 Extended Properties 4244
22.3 Custom Properties 4254
22.4 Variant Types 4257
22.5 Custom XML Data Properties 4274
22.6 Bibliography 4277
22.7 Additional Characteristics 4325

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

22.8 Office Document Relationships 4329
22.9 Shared Simple Types 4331

23. Custom XML Schema References 4354
23.1 Table of Contents 4354
23.2 Elements 4354

Annex A. (normative) Schemas – W3C XML Schema 4359
A.1 WordprocessingML 4359
A.2 SpreadsheetML 4425
A.3 PresentationML 4509
A.4 DrawingML - Framework 4540
A.5 DrawingML - Components 4605
A.6 Shared MLs 4657
A.7 Custom XML Schema References 4681

Annex B. (informative) Schemas – RELAX NG 4683
B.1 WordprocessingML 4683
B.2 SpreadsheetML 4729
B.3 PresentationML 4824
B.4 DrawingML - Framework 4850
B.5 DrawingML - Components 4899
B.6 Shared MLs 4935
B.7 Custom XML Schema References 4952
B.8 Additional Resources 4952

Annex C. (informative) Additional Syntax Constraints 4955

Annex D. (informative) Namespace Prefix Mapping in Examples 4956

Annex E. (informative) WordprocessingML Custom XML Data Extraction 4958

Annex F. (normative) WordprocessingML Page Borders 4961

Annex G. (normative) Predefined SpreadsheetML Style Definitions 4962
G.1 Built-in Table Styles 4962
G.2 Built-in Cell Styles 5018
G.3 Built-in PivotTable AutoFormats 5022

Annex H. (informative) Example Predefined DrawingML Shape and Text Geometries 5038

Annex I. (informative) Bidirectional Support 5039
I.1 Introduction 5039
I.2 Shared (WordprocessingML and DrawingML) 5039
I.3 WordprocessingML 5041
I.4 SpreadsheetML 5044
I.5 PresentationML 5045
I.6 DrawingML 5045
I.7 The Unicode Bidirectional Algorithm and Office Open XML 5045

Annex J. (informative) Accessibility Best Practices 5049
J.1 The Value of Creating an Accessible Office Open XML Implementation 5049

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

J.2 Needs by Type of Disability 5050
J.3 Best Practices for Developers 5053
J.4 Best Practices for Document and Template Authors 5056
J.5 Best Practices for Customers of Office Open XML Implementations 5069

Annex K. (informative) Root Element Locations 5082

K.1 Grouped by Part Name 5082
K.2 Grouped by Schema Name 5084

Annex L. (informative) Primer 5088

L.1 Introduction to WordprocessingML 5088
L.2 Introduction to SpreadsheetML 5180
L.3 Introduction to PresentationML 5319
L.4 Introduction to DrawingML 5351
L.5 Introduction to VML 5510
L.6 Introduction to Shared MLs 5524
L.7 Miscellaneous Topics 5555

Annex M. (informative) Differences Between ISO/IEC 29500 and ECMA-376:2006 5570

M.1 WordprocessingML 5570
M.2 SpreadsheetML 5573
M.3 PresentationML 5574
M.4 DrawingML 5575
M.5 VML 5576
M.6 Shared 5576
M.7 Custom XML Schema References 5577

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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 2.

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

This second edition cancels and replaces the first edition (ISO/IEC 29500-1:2008), which has been technically revised by incorporation of the Amendment ISO/IEC 29500-1:2008/Amd.1:2010 and the Technical Corrigendum ISO/IEC 29500-1:2008/Cor.1:2010.

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

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

Annexes A, C and D form a normative part of this Part of ISO/IEC 29500. Annexes B–9 and I–a are for information only.

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

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

Introduction

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

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

The following organizations have participated in the creation of ISO/IEC 29500 and their contributions are gratefully acknowledged:

Apple, Barclays Capital, BP, The British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba, and the United States Library of Congress

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

Part 1:
Fundamentals and Markup Language Reference

1. Scope

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

2. Conformance

The text in ISO/IEC 29500 is divided into *normative* and *informative* categories. Text marked informative (using the mechanisms described in §7) is for information purposes only. Unless stated otherwise, all text is normative.

Use of the word “shall” indicates required behavior.

Any behavior that is not explicitly specified by ISO/IEC 29500 is implicitly unspecified (§4).

2.1 Goal

This subclause is informative

The goal of this clause is to define conformance, and to provide interoperability guidelines in a way that fosters broad and innovative use of the Office Open XML file format, while maximizing interoperability and preserving investment in existing files and applications (§4). By meeting this goal, ISO/IEC 29500 benefits the following audiences:

- Developers that design, implement, or maintain Office Open XML applications.
- Developers that interact programmatically with Office Open XML applications.
- Governmental or commercial entities that procure Office Open XML applications.
- Testing organizations that verify conformance of specific Office Open XML applications to ISO/IEC 29500. (Note that ISO/IEC 29500 does not include a test suite.)
- Educators and authors who teach about Office Open XML applications.

End informative subclause

2.2 Issues

This subclause is informative

To achieve the above goal, the following issues need to be considered:

1. The application domain encompasses a range of possible consumers (§4) and producers (§4) so broad that defining specific application behaviors would restrict innovation. For example, stipulating visual layout would be inappropriate for a consumer that extracts data for machine consumption, or that renders text in sound. Another example is that restricting capacity or precision runs the risk of diluting the value of future advances in hardware.
2. Commonsense user expectations regarding the interpretation of an Office Open XML package (§4) play such an important role in that package's value that a purely syntactic definition of conformance would fail to effect a useful level of interoperability. For example, such a definition would admit an application that reads a package, and then writes it in a manner that, though syntactically valid, differs arbitrarily from the original.

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

3. Legitimate operations on a package include deliberate transformations, making blanket change prohibitions inappropriate in the conformance definition. For example, collapsing spreadsheet formulas to their calculated values, or converting complex presentation graphics to static bitmaps, could be correct for an application whose published purpose is to perform those operations. Again, commonsense user expectation makes the difference.
4. Existing files and applications exercise a broad range of formats and functionality that, if required by the conformance definition, would add an impractical amount of bulk to ISO/IEC 29500 and could inadvertently obligate new applications to implement a prohibitive amount of functionality. This issue is caused by the breadth of currently available functionality and is compounded by the existence of legacy formats.

End informative subclause

2.3 What ISO/IEC 29500 Specifies

To address the issues listed above, ISO/IEC 29500 constrains both syntax and semantics, but it is not intended to predefine application behavior. Therefore, it includes, among others, the following three types of information:

1. W3C XML Schemas and an associated validation procedure for validating document syntax against those schemas. (The validation procedure includes un-zipping, locating files, processing the extensibility XML elements and attributes, and W3C XML Schema validation.)
2. Additional syntax constraints in written form. [*Note*: These constraints are described in written form because they could not feasibly be expressed in the schema language. *end note*]
3. Descriptions of XML element semantics. The semantics of an XML element refers to its intended interpretation by a human being.

2.4 Document Conformance

Document conformance is purely syntactic; it involves only Items 1 and 2 in §2.3 above.

- A conforming document shall conform to the transitional W3C XML Schema or the strict W3C XML Schema (Item 1), and any additional syntax constraints (Item 2).
- The document shall be of category Wordprocessing, Spreadsheet, or Presentation.
- The document character set shall conform to the Unicode Standard and ISO/IEC 10646, with either the UTF-8 or UTF-16 encoding form, as required by the XML 1.0 standard.
- Any XML element or attribute not explicitly included in ISO/IEC 29500 shall use the extensibility mechanisms described by ISO/IEC 29500-1 and ISO/IEC 29500-3.

Each Part of this multi-part standard has its own conformance clause. The term *conformance class* is used to disambiguate conformance within different Parts of this multi-part standard. This Part of ISO/IEC 29500 defines the following document conformance classes:

- *WML Strict*, if the document is a conforming document of category Wordprocessing that conforms to the strict schema and does not include any features from Part 4.

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

- *SML Strict*, if the document is a conforming document of category Spreadsheet that conforms to the strict schema and does not include any features from Part 4.
- *PML Strict*, if the document is a conforming document of category Presentation that conforms to the strict schema and does not include any features from Part 4.

In addition, documents of conformance class WML Strict, SML Strict, or PML Strict shall not embed documents of conformance class WML Transitional, SML Transitional, or PML Transitional as defined in Part 4.

Document categories Wordprocessing, Spreadsheet, and Presentation are defined in §4.

[*Note*: Other document conformance classes could be defined in the future. *end note*]

[*Note*: A document cannot be of more than one of the above conformance classes. *end note*]

2.5 Application Conformance

Application conformance incorporates both syntax and semantics; it involves items 1, 2, and 3 in §2.3 above.

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

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

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

end note]

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

Conformance can also involve the use of application descriptions; see §2.6 for details.

2.6 Application Descriptions

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

The application descriptions defined within ISO/IEC 29500 are:

- Base
- Full

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

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

Each application description is identified by a URI.

The application descriptions are defined in the following subclauses.

2.6.1 Base Application Description

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

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

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

2.6.2 Full Application Description

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

2.6.3 Additional Application Descriptions

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

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

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

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

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

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

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

2.6.4 Representation of Application Descriptions within Documents

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

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

2.7 Interoperability Guidelines

[*Guidance*: The following interoperability guidelines incorporate semantics (Item 3 in §2.3 above).

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

The following scenarios illustrate these guidelines.

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

The normative requirements in §2.4 imply that a conforming producer shall not write unescaped non-XML characters. As an implementation guideline, a conforming producer additionally should not write escaped non-XML characters. Doing so damages interoperability with existing XML-based standards such as SOAP and RDF. For example, implementers could either refuse to create documents including such characters, or warn users that including such characters compromises the re-usability of their documents. *end guidance*]

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

3. Normative References

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

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

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

Calendar Reform Committee, Indian Ephemeris and Nautical Almanac. 1957

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

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

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

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

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

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

Korean Law Enactment No. 4, 1961.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

4. Terms and Definitions

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

application — A consumer or producer.

behavior — External appearance or action.

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

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

behavior, unspecified —Behavior where ISO/IEC 29500 makes no recommendations. [*Note*: To add an extension, an implementer must use the extensibility mechanisms described by ISO/IEC 29500 rather than trying to do so by giving meaning to otherwise unspecified behavior. *end note*]

comment — A note that an author or reviewer attaches to content in a document. Although a consumer might chose to display comments, they are not considered part of the body of the document. A comment might include the text of the note, the comment author's name and initials, and date of creation, among other things.

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

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

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

- A document whose package-relationship item contains a relationship to a Main Document part (§11.3.10) is a document of category Wordprocessing.
- A document whose package-relationship item contains a relationship to a Workbook part (§12.3.23) is a document of category Spreadsheet.

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

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

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

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

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

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

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

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

relationship, implicit — A relationship that is not explicit.

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

5. Notational Conventions

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

6. Acronyms and Abbreviations

This clause is informative

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

IEC — the International Electrotechnical Commission

ISO — the International Organization for Standardization

W3C — World Wide Web Consortium

End of informative text

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

7. General Description

ISO/IEC 29500 is intended for use by implementers, academics, and application programmers. As such, it contains a considerable amount of explanatory material that, strictly speaking, is not necessary in a formal specification.

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

8. Overview

This clause is informative.

This clause contains an overview of Office Open XML.

8.1 Packages and Parts

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

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

8.2 Consumers and Producers

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

8.3 WordprocessingML

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

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

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

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

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

8.4 SpreadsheetML

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

8.5 PresentationML

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

8.6 Supporting MLs

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

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

8.6.1 DrawingML

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

8.6.2 Custom XML Data Properties

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

8.6.3 File Properties

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

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

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

8.6.4 Math

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

8.6.5 Bibliography

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

End of informative text.

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

9. Packages

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

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

9.1 Office Open XML's Use of OPC

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

9.1.1 Part Addressing

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

9.1.2 Fragments

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

9.1.3 Physical Packages

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

9.1.4 Unknown Parts

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

9.1.5 Trash Items

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

end example]

9.1.6 Invalid Parts

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

9.1.7 Unknown Relationships

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

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

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

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

9.2 Relationships in Office Open XML

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

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

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

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

note]. Relationships that are required or permitted, and restrictions on those relationships are described in §10–15 of this Part of ISO/IEC 29500.

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

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

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

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

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

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

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

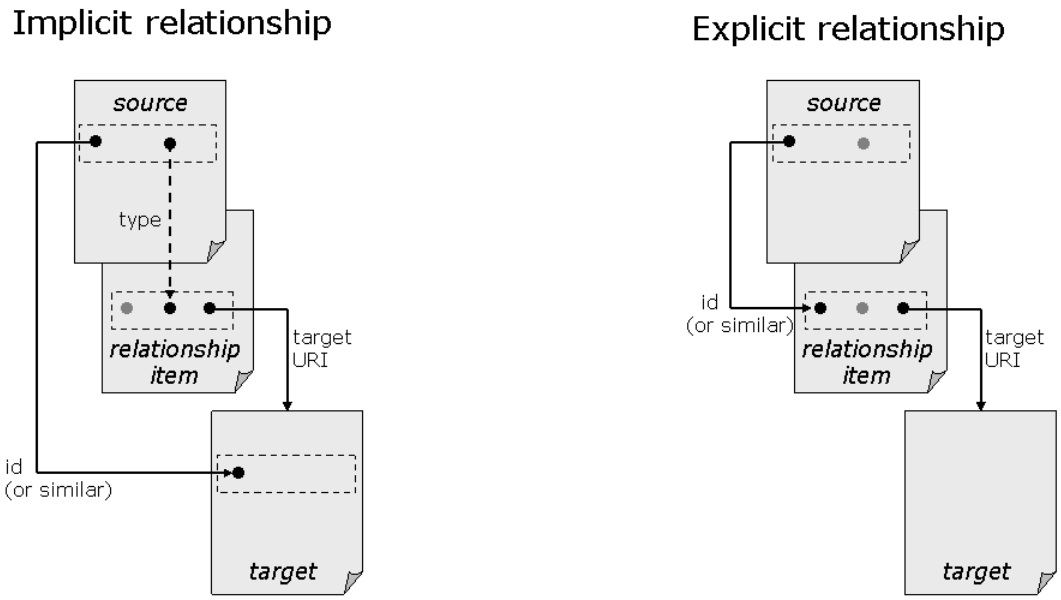
The dots correspond to attributes of relevant elements. Where one attribute refers to a piece in another part, this is indicated by arrows. Solid arrows indicate that the value of the source directly corresponds to the value at the target (for instance, id=rId4 in the source part corresponds to id=rId4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

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

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

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

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

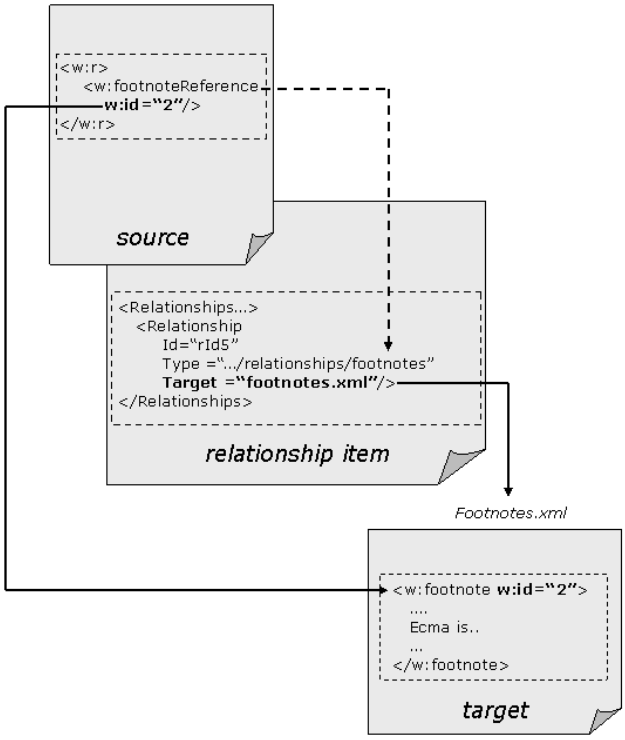


end example]

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

Example Implicit relationship



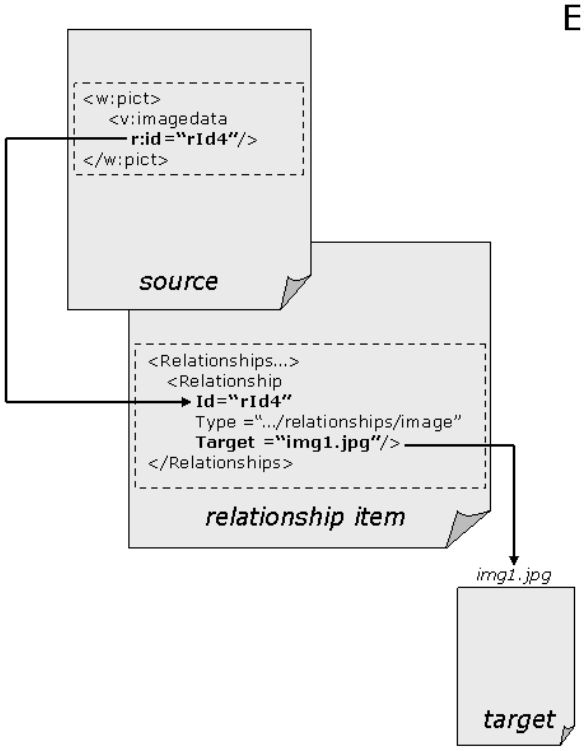
end example]

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

Example Explicit relationship



end example]

©ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

10. Markup Compatibility and Extensibility

Office Open XML documents are designed to allow for innovation by extending their capabilities via a scheme defined by Part 3. This subclause contains information regarding Office Open XML's use of the Markup Compatibility constructs.

10.1 Constraints on Office Open XML's Use of Markup Compatibility and Extensibility

While the Markup Compatibility and Extensibility specification is designed for and used by Office Open XML documents, it could also be used to support a much broader range of applications. As a result, the use of some Markup Compatibility and Extensibility features is restricted within Office Open XML documents. These additional requirements are discussed in the following subordinate subclauses. Unless explicitly specified below, all normative requirements of the Markup Compatibility and Extensibility specification apply to Office Open XML documents.

10.1.1 PreserveElements and PreserveAttributes

The PreserveElements and PreserveAttributes elements, as defined in Part 5, allow a markup language to specify the conditions under which extensions should be round-tripped, even when their contents are edited. Within the context of the markup languages explicitly defined by ISO/IEC 29500, no such conditions are specified, and therefore applications are not obliged to support these hints at any point in an Office Open XML document. Instead, the well-defined extensibility constructs defined below should be used.

All other constructs defined in Part 5 shall be supported.

10.1.2 Office Open XML Native Extensibility Constructs

Clause 12 of Part 3 specifies the ability for a markup language to define additional constructs for extensibility of a specific markup language. Within the context of Office Open XML documents, the extLst element(s) defined in individual markup languages shall allow the round-tripping of all unknown content regardless of the state of the PreserveElements and PreserveAttributes elements. See the reference material in §17–23 for additional information on the XML elements that allow the use of the extLst construct.

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

11. WordprocessingML

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

11.1 Glossary of WordprocessingML-Specific Terms

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

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

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

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

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

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

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

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

11.2 Package Structure

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

- One Main Document part (§11.3.10)

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

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

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

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

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

end example]

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

The package-relationship item contains the following:

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

end example]

11.3 Part Summary

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

Part	Relationship Target of	Root Element	Ref.
Alternative Format	Comments, Endnotes, Footer,	Not applicable	§11.3.1

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

Part	Relationship Target of	Root Element	Ref.
Import	Footnotes, Header, or Main Document		
Comments	Glossary Document or Main Document	comments	§11.3.2
Document Settings	Glossary Document or Main Document	settings	§11.3.3
Endnotes	Glossary Document or Main Document	endnotes	§11.3.4
Font Table	Glossary Document or Main Document	fonts	§11.3.5
Footer	Glossary Document or Main Document	ftr	§11.3.6
Footnotes	Glossary Document or Main Document	footnotes	§11.3.7
Glossary Document	Main Document	glossaryDocument	§11.3.8
Header	Glossary Document or Main Document	hdr	§11.3.9
Main Document	WordprocessingML package	document	§11.3.10
Numbering Definitions	Glossary Document or Main Document	numbering	§11.3.11
Style Definitions	Glossary Document or Main Document	styles	§11.3.12
Web Settings	Glossary Document or Main Document	webSettings	§11.3.13

end note]

11.3.1 Alternative Format Import Part

Content Type:	One of the following formats: • Text = text/plain • HTML = text/html • WordprocessingML = application/vnd.openxmlformats-officedocument.wordprocessingml.document • XHTML = application/xhtml+xml
Root Namespace:	not applicable
Source Relationship:	http://purl.oclc.org/ooxml/officeDocument/relationships/aFChunk

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

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

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

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

The Main Document part contains the following XML fragment:

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

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

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

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

11.3.2 Comments Part

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

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

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

end example]

The root element for a Comment part shall be comments.

[*Example*:

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

end example]

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

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

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

end example]

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

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

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

end example]

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

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

11.3.3 Document Settings Part

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

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

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

[*Example*: The following Main Document part-relationship item contains a relationship to a Document Settings part, which is stored in the ZIP item documentProperties1.xml:

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

end example]

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

[*Example*:

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

end example]

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

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

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

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

11.3.4 Endnotes Part

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

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

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub

Full document details: ISO_IEC 29500-1_2011

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

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

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

end example]

The root element for an Endnotes part shall be endnotes.

[*Example*:

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

end example]

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

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

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

©ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011

end example]

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

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

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

end example]

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

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

- Alternative Format Import (§11.3.1)
- Chart (§14.2.1)
- Content Part (§15.2.4)

©ISO/IEC 2011 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO_IEC 29500-1_2011