INTERNATIONAL **ISO/IEC**

STANDARD **29500-1**

Fourth edition

2016-11-01

**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 29500-1:2016(E)

© ISO/IEC 2016

**ISO/IEC 29500-1:2016(E)**

 **COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2016, Published in Switzerland

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

ISO copyright office

Ch. de Blandonnet 8 • CP 401

CH-1214 Vernier, Geneva, Switzerland

Tel. +41 22 749 01 11 Fax +41 22 749 09 47 copyright@iso.org www.iso.org

ii© ISO/IEC 2016 – All rights reserved

**Table of Contents**

[Foreword viii](#_Toc10426541)

[Introduction x](#_Toc10426542)

[1. Scope 1](#_Toc10426543)

[2. Conformance 2](#_Toc10426544)

[2.1 Document Conformance 2](#_Toc10426545)

[2.2 Application Conformance 2](#_Toc10426546)

[2.3 Application Descriptions 3](#_Toc10426547)

[2.4 Interoperability Guidelines 5](#_Toc10426548)

[3. Normative References 6](#_Toc10426549)

[4. Terms and Definitions 10](#_Toc10426550)

[5. Notational Conventions 13](#_Toc10426551)

[6. Acronyms and Abbreviations 14](#_Toc10426552)

[7. General Description 15](#_Toc10426553)

[8. Overview 16](#_Toc10426554)

[8.1 Content Overview 16](#_Toc10426555)

[8.2 Packages and Parts 16](#_Toc10426556)

[8.3 Consumers and Producers 16](#_Toc10426557)

[8.4 WordprocessingML 16](#_Toc10426558)

[8.5 SpreadsheetML 18](#_Toc10426559)

[8.6 PresentationML 18](#_Toc10426560)

[8.7 Supporting MLs 19](#_Toc10426561)

[9. Packages 21](#_Toc10426562)

[9.1 Office Open XML's Use of OPC 21](#_Toc10426563)

[9.2 Relationships in Office Open XML 22](#_Toc10426564)

[10. Markup Compatibility and Extensibility 27](#_Toc10426565)

[11. WordprocessingML 28](#_Toc10426566)

[11.1 Glossary of WordprocessingML-Specific Terms 28](#_Toc10426567)

[11.2 Package Structure 28](#_Toc10426568)

[11.3 Part Summary 31](#_Toc10426569)

[11.4 Document Template 57](#_Toc10426570)

[11.5 Framesets 58](#_Toc10426571)

[11.6 Master Documents and Subdocuments 59](#_Toc10426572)

[11.7 Mail Merge Data Source 60](#_Toc10426573)

[11.8 Mail Merge Header Data Source 61](#_Toc10426574)

[11.9 XSL Transformation 62](#_Toc10426575)

[12. SpreadsheetML 64](#_Toc10426576)

[12.1 Glossary of SpreadsheetML-Specific Terms 64](#_Toc10426577)

[12.2 Package Structure 65](#_Toc10426578)

[12.3 Part Summary 67](#_Toc10426579)

[12.4 External Workbooks 102](#_Toc10426580)

[13. PresentationML 103](#_Toc10426581)

[13.1 Glossary of PresentationML-Specific Terms 103](#_Toc10426582)

[13.2 Package Structure 103](#_Toc10426583)

[13.3 Part Summary 106](#_Toc10426584)

[13.4 HTML Publish Location 124](#_Toc10426585)

[13.5 Slide Synchronization Server Location 125](#_Toc10426586)

[14. DrawingML 127](#_Toc10426587)

[14.1 Glossary of DrawingML-Specific Terms 127](#_Toc10426588)

[14.2 Part Summary 127](#_Toc10426589)

[15. Shared 140](#_Toc10426590)

[15.1 Glossary of Shared Terms 140](#_Toc10426591)

[15.2 Part Summary 141](#_Toc10426592)

[15.3 Hyperlinks 162](#_Toc10426593)

[16. Part Overview 164](#_Toc10426594)

[16.1 WordprocessingML Summary 164](#_Toc10426595)

[16.2 SpreadsheetML Summary 164](#_Toc10426596)

[16.3 PresentationML Summary 165](#_Toc10426597)

[16.4 DrawingML Summary 166](#_Toc10426598)

[16.5 Shared Summary 166](#_Toc10426599)

[17. WordprocessingML Reference Material 169](#_Toc10426600)

[17.1 Table of Contents 169](#_Toc10426601)

[17.2 Main Document Story 189](#_Toc10426602)

[17.3 Paragraphs and Rich Formatting 195](#_Toc10426603)

[17.4 Tables 373](#_Toc10426604)

[17.5 Custom Markup 485](#_Toc10426605)

[17.6 Sections 547](#_Toc10426606)

[17.7 Styles 614](#_Toc10426607)

[17.8 Fonts 670](#_Toc10426608)

[17.9 Numbering 692](#_Toc10426609)

[17.10 Headers and Footers 734](#_Toc10426610)

[17.11 Footnotes and Endnotes 747](#_Toc10426611)

[17.12 Glossary Document 780](#_Toc10426612)

[17.13 Annotations 798](#_Toc10426613)

[17.14 Mail Merge 929](#_Toc10426614)

[17.15 Settings 970](#_Toc10426615)

[17.16 Fields and Hyperlinks 1158](#_Toc10426616)

[17.17 Miscellaneous Topics 1293](#_Toc10426617)

[17.18 Simple Types 1302](#_Toc10426618)

[18. SpreadsheetML Reference Material 1523](#_Toc10426619)

[18.1 Table of Contents 1523](#_Toc10426620)

[18.2 Workbook 1542](#_Toc10426621)

[18.3 Worksheets 1589](#_Toc10426622)

[18.4 Shared String Table 1717](#_Toc10426623)

[18.5 Tables 1726](#_Toc10426624)

[18.6 Calculation Chain 1742](#_Toc10426625)

[18.7 Comments 1745](#_Toc10426626)

[18.8 Styles 1752](#_Toc10426627)

[18.9 Metadata 1801](#_Toc10426628)

[18.10 Pivot Tables 1815](#_Toc10426629)

[18.11 Shared Workbook Data 1959](#_Toc10426630)

[18.12 QueryTable Data 1988](#_Toc10426631)

[18.13 External Data Connections 1995](#_Toc10426632)

[18.14 Supplementary Workbook Data 2016](#_Toc10426633)

[18.15 Volatile Dependencies 2026](#_Toc10426634)

[18.16 Custom XML Mappings 2031](#_Toc10426635)

[18.17 Formulas 2039](#_Toc10426636)

[18.18 Simple Types 2434](#_Toc10426637)

[19. PresentationML Reference Material 2516](#_Toc10426638)

[19.1 Table of Contents 2516](#_Toc10426639)

[19.2 Presentation 2522](#_Toc10426640)

[19.3 Slides 2559](#_Toc10426641)

[19.4 Comments 2597](#_Toc10426642)

[19.5 Animation 2601](#_Toc10426643)

[19.6 Slide Synchronization Data 2690](#_Toc10426644)

[19.7 Simple Types 2691](#_Toc10426645)

[20. DrawingML - Framework Reference Material 2719](#_Toc10426646)

[20.1 DrawingML - Main 2719](#_Toc10426647)

[20.2 DrawingML - Picture 3087](#_Toc10426648)

[20.3 DrawingML - Locked Canvas 3095](#_Toc10426649)

[20.4 DrawingML - WordprocessingML Drawing 3096](#_Toc10426650)

[20.5 DrawingML - SpreadsheetML Drawing 3152](#_Toc10426651)

[21. DrawingML - Components Reference Material 3181](#_Toc10426652)

[21.1 DrawingML - Main 3181](#_Toc10426653)

[21.2 DrawingML - Charts 3361](#_Toc10426654)

[21.3 DrawingML - Chart Drawings 3469](#_Toc10426655)

[21.4 DrawingML - Diagrams 3490](#_Toc10426656)

[22. Shared MLs Reference Material 3599](#_Toc10426657)

[22.1 Math 3599](#_Toc10426658)

[22.2 Extended Properties 3720](#_Toc10426659)

[22.3 Custom Properties 3727](#_Toc10426660)

[22.4 Variant Types 3729](#_Toc10426661)

[22.5 Custom XML Data Properties 3740](#_Toc10426662)

[22.6 Bibliography 3743](#_Toc10426663)

[22.7 Additional Characteristics 3780](#_Toc10426664)

[22.8 Office Document Relationships 3784](#_Toc10426665)

[22.9 Shared Simple Types 3785](#_Toc10426666)

[23. Custom XML Schema References 3800](#_Toc10426667)

[23.1 Table of Contents 3800](#_Toc10426668)

[23.2 Elements 3800](#_Toc10426669)

[Annex A. (normative) Schemas – W3C XML Schema 3805](#_Toc10426670)

[A.1 WordprocessingML 3805](#_Toc10426671)

[A.2 SpreadsheetML 3871](#_Toc10426672)

[A.3 PresentationML 3955](#_Toc10426673)

[A.4 DrawingML - Framework 3986](#_Toc10426674)

[A.5 DrawingML - Components 4054](#_Toc10426675)

[A.6 Shared MLs 4105](#_Toc10426676)

[A.7 Custom XML Schema References 4129](#_Toc10426677)

[Annex B. (informative) Schemas – RELAX NG 4131](#_Toc10426678)

[B.1 WordprocessingML 4132](#_Toc10426679)

[B.2 SpreadsheetML 4178](#_Toc10426680)

[B.3 PresentationML 4273](#_Toc10426681)

[B.4 DrawingML - Framework 4298](#_Toc10426682)

[B.5 DrawingML - Components 4350](#_Toc10426683)

[B.6 Shared MLs 4386](#_Toc10426684)

[B.7 Custom XML Schema References 4402](#_Toc10426685)

[B.8 Additional Resources 4403](#_Toc10426686)

[Annex C. (informative) Additional Syntax Constraints 4405](#_Toc10426687)

[Annex D. (informative) Namespace Prefix Mapping in Examples 4406](#_Toc10426688)

[Annex E. (informative) WordprocessingML Custom XML Data Extraction 4408](#_Toc10426689)

[Annex F. (normative) WordprocessingML Page Borders 4411](#_Toc10426690)

[Annex G. (normative) Predefined SpreadsheetML Style Definitions 4412](#_Toc10426691)

[G.1 Built-in Table Styles 4412](#_Toc10426692)

[G.2 Built-in Cell Styles 4468](#_Toc10426693)

[G.3 Built-in PivotTable AutoFormats 4472](#_Toc10426694)

[Annex H. (informative) Example Predefined DrawingML Shape and Text Geometries 4488](#_Toc10426695)

[Annex I. (informative) Bidirectional Support 4489](#_Toc10426696)

[I.1 Introduction 4489](#_Toc10426697)

[I.2 Shared (WordprocessingML and DrawingML) 4489](#_Toc10426698)

[I.3 WordprocessingML 4491](#_Toc10426699)

[I.4 SpreadsheetML 4494](#_Toc10426700)

[I.5 PresentationML 4495](#_Toc10426701)

[I.6 DrawingML 4495](#_Toc10426702)

[I.7 The Unicode Bidirectional Algorithm and Office Open XML 4495](#_Toc10426703)

[Annex J. (informative) Accessibility Best Practices 4499](#_Toc10426704)

[J.1 The Value of Creating an Accessible Office Open XML Implementation 4499](#_Toc10426705)

[J.2 Needs by Type of Disability 4500](#_Toc10426706)

[J.3 Best Practices for Developers 4503](#_Toc10426707)

[J.4 Best Practices for Document and Template Authors 4506](#_Toc10426708)

[J.5 Best Practices for Customers of Office Open XML Implementations 4519](#_Toc10426709)

[Annex K. (informative) Root Element Locations 4532](#_Toc10426710)

[K.1 Grouped by Part Name 4532](#_Toc10426711)

[K.2 Grouped by Schema Name 4534](#_Toc10426712)

[Annex L. (informative) Primer 4538](#_Toc10426713)

[L.1 Introduction to WordprocessingML 4538](#_Toc10426714)

[L.2 Introduction to SpreadsheetML 4630](#_Toc10426715)

[L.3 Introduction to PresentationML 4769](#_Toc10426716)

[L.4 Introduction to DrawingML 4801](#_Toc10426717)

[L.5 Introduction to VML 4960](#_Toc10426718)

[L.6 Introduction to Shared MLs 4974](#_Toc10426719)

[L.7 Miscellaneous Topics 5005](#_Toc10426720)

[Annex M. (informative) Differences Between ISO/IEC 29500 and ECMA-376:2006 5016](#_Toc10426721)

[M.1 WordprocessingML 5016](#_Toc10426722)

[M.2 SpreadsheetML 5019](#_Toc10426723)

[M.3 PresentationML 5020](#_Toc10426724)

[M.4 DrawingML 5021](#_Toc10426725)

[M.5 VML 5022](#_Toc10426726)

[M.6 Shared 5022](#_Toc10426727)

[M.7 Custom XML Schema References 5023](#_Toc10426728)

[Bibliography 5024](#_Toc10426729)

# 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 fourth edition cancels and replaces the third edition (ISO/IEC 29500-1:2012), which has been technically revised by incorporation of the Technical Corrigenda ISO/IEC 29500-1:2012/Cor.1:2015 and ISO/IEC 295001:2012/Cor.2:2016.

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

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

# 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.

**INTERNATIONAL STANDARD ISO/IEC 29500-1:2016(E)**

**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 preexisting 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.

1

# 2. Conformance

## 2.1 Document Conformance

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

* The document obeys all constraints specified in this Part of ISO/IEC 29500
* The document is of category Wordprocessing, Spreadsheet, or Presentation, as defined in §4
* For each OPC Part of the document of the types listed in §11.3, §12.3, §13.3, §14.2 or §15.2, all the following shall hold:

1. The Part may contain markup in the Markup Compatibility namespace as specified in ISO/IEC 29500-3
2. After the removal of any extensions by an MCE processor as specified in ISO/IEC 29500-3, the part is valid against the strict W3C XML Schema (Appendix A)

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

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

## 2.2 Application Conformance

Application conformance incorporates both syntax and semantics:

* 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:

* *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 §2.3 for details.

## 2.3 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.

#### 2.3.1 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*]

#### 2.3.2 Full Application Description

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

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

#### 2.3.3 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*]

#### 2.3.4 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*]

## 2.4 Interoperability Guidelines

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

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 §2.1 imply that a conforming producer shall not write unescaped non-XML characters. As an implementation guideline, a conforming producer additionally should not write escaped nonXML 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*]

# 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*. Department of Mathematics, Technion, Israel Institute 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/mediatypes/](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 9075-3, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)*

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

*Interfaces*

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

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

*ISO/IEC 29500-3:2015, Information technology — Document description and processing languages — Office*

*Open XML File Formats – Part 3: Markup Compatibility and Extensibility*

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.](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.](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

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](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.](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 92-9502537-7, 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/RECxmlbase-20010627/

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-xmlnames-20091208/](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.](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

# 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. (This term is sometimes called “application-dependent behavior”.) [*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*] **byte** — A sequence of 8 bits treated as a unit.

**comment** — A note attached to content in a document. Although a consumer might choose 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.
* 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.

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

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

**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.

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

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

# 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 a distinct style and typeface. [*Example*: The bookmarkStart and bookmarkEnd elements specify … *end example*]
* The name of an XML attribute is written using a distinct style and typeface. [*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 distinct style and typeface. [*Example*: The possible values for this attribute are defined by the ST\_HexColor simple type. *end example*]

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

# 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**

# 7. General Description

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.**"

# 8. Overview

**This clause is informative.**

This clause contains an overview of Office Open XML.

## 8.1 Content Overview

This standard contains predominantly the following three types of information:

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

## 8.2 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 *partrelationship 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.

## 8.3 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.

## 8.4 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.

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 WordprocessingML document 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.

## 8.5 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*.

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.

## 8.6 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.

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.

## 8.7 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.

#### 8.7.1 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.

#### 8.7.2 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.

#### 8.7.3 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.

#### 8.7.4 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

#### 8.7.5 Bibliography

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

**End of informative text.**

# 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

*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*]

**9.1.8 Interleaving**

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

## 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 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 Ecma1 (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.

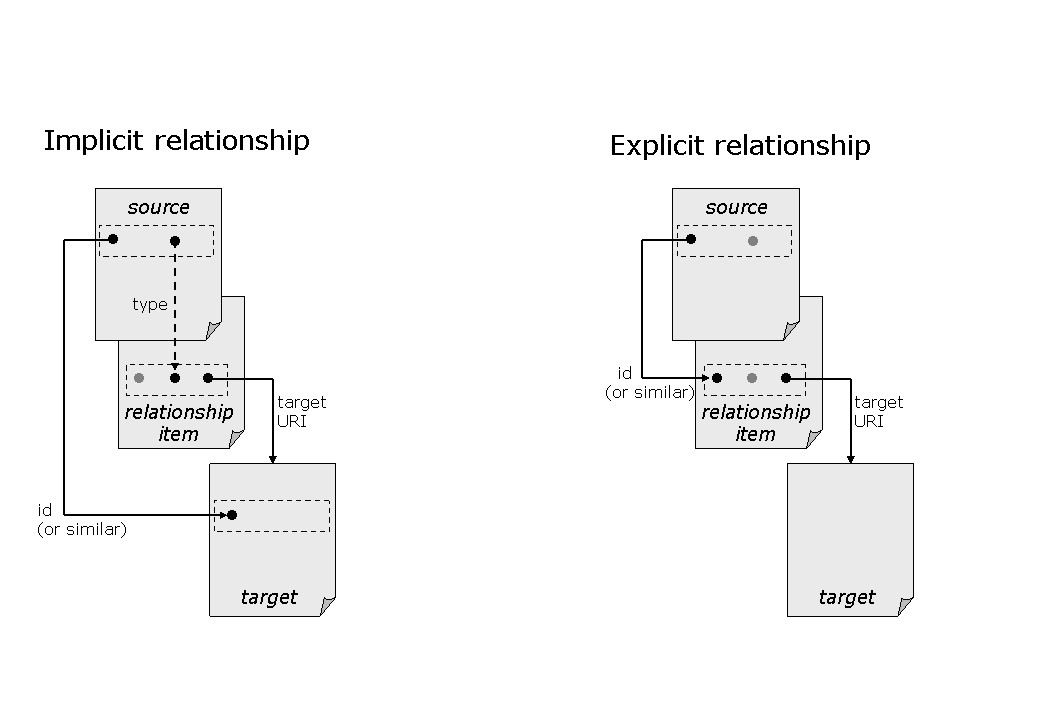
The dots correspond to attributes of relevant elements. Where one attribute refers to a piece in another part, this is indicated by arrows. Solid arrows indicate that the value of the source directly corresponds to the value at the target (for instance, id=rId4 in the source part corresponds to id=rId4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

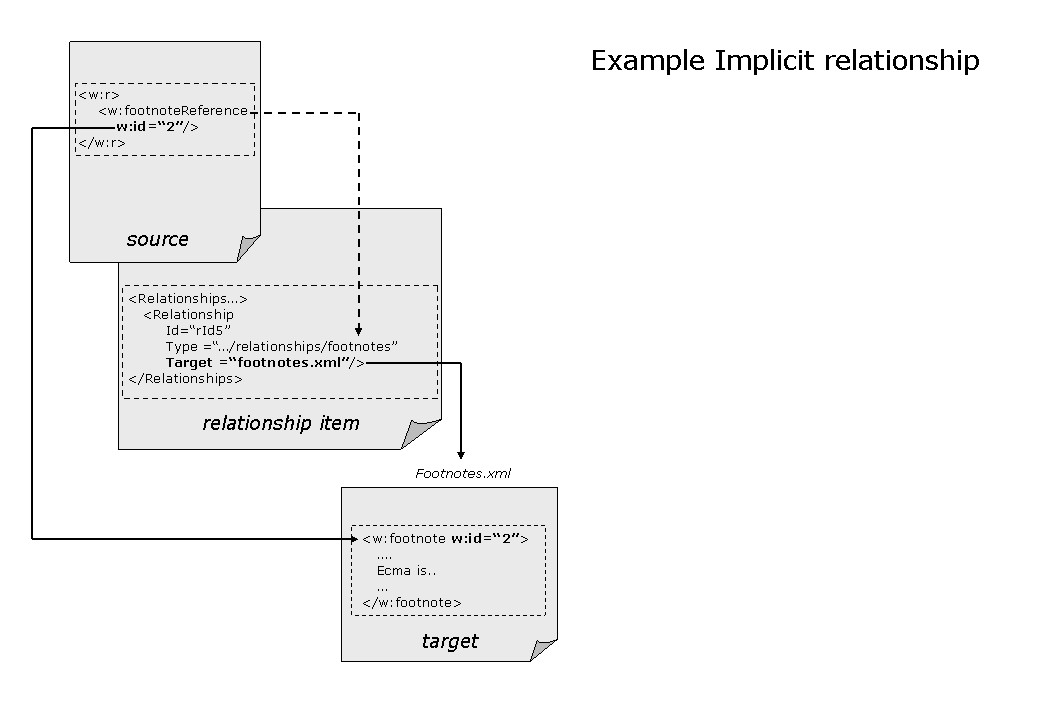
Both relationship types use the target URI of the relationship in the relationship item to locate the target.

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.



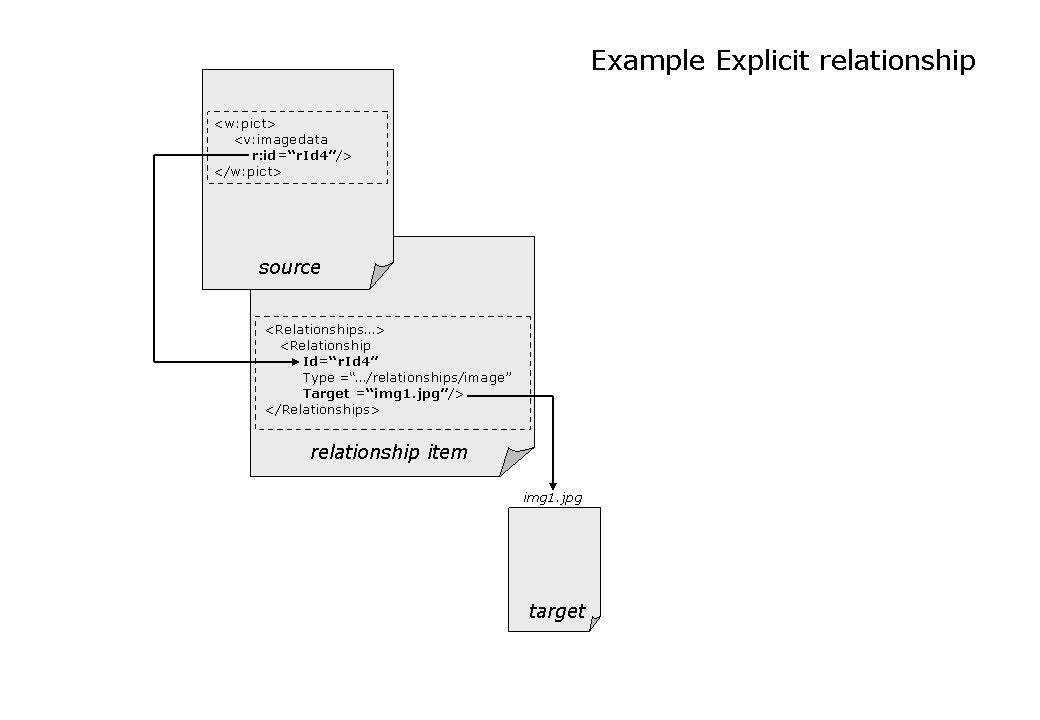
*end example*]

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



*end example*]

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



*end example*]

# 10. Markup Compatibility and Extensibility

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

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

**document building block** — 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 packagerelationship 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:

* 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:

|  |  |  |  |
| --- | --- | --- | --- |
| /[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  /word/footer2.xml  /word/footer3.xml  /word/footer4.xml |  |  | *Footer parts* |
| /word/footnotes.xml |  |  | *Footnotes part* |
| /word/header1.xml  /word/header2.xml  /word/header3.xml  /word/header4.xml  /word/header5.xml  /word/header6.xml |  |  | *Header parts* |
| /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 Import | Comments, Endnotes, Footer,  Footnotes, Header, or Main  Document | Not applicable | §11.3.1 |
| 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: | Any text-based content, support for which is application-defined. [*Note*: Some examples of formats which might be supported include:     * Text = text/plain * HTML = text/html * WordprocessingML = application/vnd.openxmlformats-officedocument.wordprocessingml.document  XHTML = application/xhtml+xml |
|  | *end note*] |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/aFChunk |

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:

<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>

*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*]

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 settings.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.

[*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.

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>

*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: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)
* 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)

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

#### 11.3.5 Font Table Part

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

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

A package shall contain no more than two Font Table parts. If it exists, one instance of that part shall be the target of an implicit relationship in the part-relationship item for the Main Document (§11.3.10) part, and the other instance 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 Font Table part, which is stored as the ZIP item fontTable.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/fontTable" Target="fontTable.xml"/> </Relationships>

*end example*]

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

[*Example*:

<w:fonts … >

<w:font w:name="Calibri">

<w:panose1 w:val="020F0502020204030204"/>

<w:charset w:val="00"/>

<w:family w:val="swiss"/>

<w:pitch w:val="variable"/>

<w:sig w:usb0="A00002EF" w:usb1="4000207B" w:usb2="00000000" w:usb3="00000000" w:csb0="0000009F" w:csb1="00000000"/>

</w:font> </w:fonts>

*end example*]

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

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

 Fonts (§15.2.13)

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

#### 11.3.6 Footer Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId91"

Type="http://…/footer" Target="footer3.xml"/> </Relationships>

*end example*]

The root element for a Footer part shall be ftr.

[*Example*:

<w:ftr xmlns:w="…" …>

…

</w:ftr>

*end example*]

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

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

<w:document xmlns:w="…">

…

<w:sectPr>

<w:footerReference w:val="rId89" w:type="default"/>

<w:footerReference w:val="rId90" w:type="even"/>

<w:footerReference w:val="rId91" w:type="first"/>

<w:type w:val="oddPage"/>

<w:pgSz w:w="11909" w:h="16834" w:code="9"/> <w:pgMar w:top="1440" w:right="1152" w:bottom="1440" w:left="1152" w:header="720" w:footer="720" w:gutter="0"/>

<w:lnNumType w:countBy="1"/>

<w:pgNumType w:numFmt="lowerRoman"/>

<w:cols w:space="720"/>

</w:sectPr>

</w:document>

*end example*]

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

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

<w:ftr xmlns:w="…">

<w:p>

<w:pPr>

<w:pStyle w:val="Centered"/>

</w:pPr>

<w:fldSimple w:instr="PAGE">

<w:r>

<w:rPr>

<w:noProof/>

</w:rPr>

<w:t>i</w:t>

</w:r>

</w:fldSimple>

</w:p>

</w:ftr>

*end example*]

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

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

#### 11.3.7 Footnotes Part

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

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

A package shall contain no more than two Footnotes 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 Main Document part-relationship item contains a relationship to the Footnotes part, which is stored as the ZIP item footnotes.xml:

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/footnotes" Target="footnotes.xml"/> </Relationships>

*end example*]

The root element for a Footnotes part shall be footnotes.

[*Example*:

<w:footnotes xmlns:w="…" …>

<w:footnote …>

…

</w:footnote>

<w:footnote …>

…

</w:footnote> </w:footnotes>

*end example*]

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

[*Example*: Consider the case in which the Main Document part contains the text "… in the Standard.", and there is a footnote 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="FootnoteReference"/>

</w:rPr>

<w:footnoteReference w:id="5"/>

</w:r> </w:p>

*end example*]

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

[*Example*: The text of the footnote 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:footnotes xmlns:w="…"

<w:footnote w:id="5">

<w:p>

<w:r>

<w:rPr>

<w:rStyle w:val="FootnoteReference"/>

</w:rPr>

<w:footnoteRef/>

</w:r>

<w:r>

<w:t xml:space="preserve">This can be downloaded from </w:t> </w:r>

<w:hyperlink r:id="rId2" w:history="1">

<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:r>

</w:p>

</w:footnote> </w:footnotes>

*end example*]

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

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

#### 11.3.8 Glossary Document Part

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

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

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

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

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

<w:glossaryDocument xmlns:w="…" >

<w:docParts>

<w:docPart>

<w:docPartPr>

<w:name w:val="rainbow colors"/>

<w:style w:val="Normal"/>

<w:category>

<w:name w:val="Misc"/>

<w:gallery w:val="docParts"/>

</w:category>

</w:docPartPr>

<w:docPartBody>

<w:p>

<w:r>

<w:t>The colors of the rainbow are red, orange, yellow, green, blue, indigo, and violet.</w:t>

</w:r>

</w:p>

</w:docPartBody>

</w:docPart>

<w:docPart>

…

</w:docPart>

</w:docParts>

</w:glossaryDocument>

*end example*]

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

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

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/glossaryDocument" Target="glossary/document.xml"/> </Relationships>

*end example*]

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

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

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

A Glossary Document part is permitted to have 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)
* Footer (§11.3.6)
* Header (§11.3.9)
* Hyperlinks (§15.3)
* Images (§15.2.14)
* Printer Settings (§15.2.15)
* Video (§15.2.17)

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

#### 11.3.9 Header Part

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

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

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

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

<Relationships xmlns="…">

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

*end example*]

The root element for a Header part shall be hdr.

[*Example*:

<w:hdr xmlns:w="…" …>

…

</w:hdr>

*end example*]

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

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

<w:body>

…

<w:sectPr w:rsidR="00363F31" w:rsidSect="008D4B40">

<w:headerReference w:val="rId89" w:type="default"/>

<w:headerReference w:val="rId90" w:type="even"/>

<w:headerReference w:val="rId91" w:type="first"/>

<w:type w:val="oddPage"/>

<w:pgSz w:w="11909" w:h="16834" w:code="9"/> <w:pgMar w:top="1440" w:right="1152" w:bottom="1440" w:left="1152" w:header="720" w:footer="720" w:gutter="0"/>

<w:lnNumType w:countBy="1"/>

<w:pgNumType w:fmt="lowerRoman"/>

<w:cols w:space="720"/>

</w:sectPr> </w:body>

*end example*]

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

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

<w:hdr xmlns:w="…">

<w:p>

<w:pPr>

<w:pStyle w:val="Header"/>

</w:pPr>

<w:r>

<w:t>My Test Document</w:t>

</w:r>

</w:p>

</w:hdr>

Here is the odd header corresponding to the examples above:

<w:hdr xmlns:w="…">

<w:p>

<w:pPr>

<w:pStyle w:val="Header"/>

</w:pPr>

<w:r>

<w:tab/>

<w:t>Table of Contents</w:t>

</w:r>

</w:p>

</w:hdr>

*end example*]

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

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

#### 11.3.10 Main Document Part

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

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/officeDocument" Target="word/document.xml"/> </Relationships>

/word/document.xml" contains the following:

<w:document …>

<w:body>

…

</w:body>

</w:document>

*end example*]

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

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

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

A Main Document 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)
* Footer (§11.3.6)
* Header (§11.3.9)
* Hyperlinks (§15.3)
* Images (§15.2.14)
* Printer Settings (§15.2.15)
* Subdocument (§11.6)  Video (§15.2.17)

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

#### 11.3.11 Numbering Definitions Part

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

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

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

 First level

 Second level

 Third level

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

A package shall contain no more than two Numbering Definitions parts. If they exist, 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*:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/numbering" Target="numbering.xml"/> </Relationships>

*end example*]

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

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

<w:p>

<w:pPr>

<w:pStyle w:val="Text"/>

</w:pPr>

<w:r>

<w:t>The kinds of fruit needed are:</w:t>

</w:r>

</w:p>

<w:p>

<w:pPr>

<w:pStyle w:val="ListBullet"/>

<w:numPr>

<w:ilvl w:val="0" />

<w:numId w:val="5" />

</w:numPr>

</w:pPr>

<w:r>

<w:t>Apples</w:t>

</w:r>

</w:p>

<w:p>

<w:pPr>

<w:pStyle w:val="ListBullet"/>

<w:numPr>

<w:ilvl w:val="0" />

<w:numId w:val="5" />

</w:numPr>

</w:pPr>

<w:r>

<w:t>Oranges</w:t>

</w:r>

</w:p>

<w:p>

<w:pPr>

<w:pStyle w:val="Text"/>

</w:pPr>

<w:r>

<w:t>Other items may be needed too.</w:t>

</w:r> </w:p>

*end example*]

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

[*Example*:

<w:numbering xmlns:w="…">

<w:abstractNum w:numId="11">

<w:nsid w:val="394E2425"/>

<w:multiLevelType w:val="hybridMultilevel"/>

<w:tmpl w:val="F628E89A"/>

<w:lvl w:ilvl="0" w:tplc="151C4798">

<w:start w:val="1"/>

<w:numFmt w:val="bullet"/>

<w:pStyle w:val="ListBullet"/>

<w:lvlText w:val="…"/>

<w:lvlJc w:val="start"/>

<w:pPr>

<w:tabs>

<w:tab w:val="num" w:pos="720"/>

</w:tabs>

<w:ind w:start="720" w:hanging="360"/> </w:pPr>

<w:rPr>

<w:rFonts w:ascii="Symbol" w:hAnsi="Symbol" w:hint="default"/>

</w:rPr>

</w:lvl>

…

</w:abstractNum> </w:numbering>

*end example*]

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

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

 Images (§15.2.14)

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

#### 11.3.12 Style Definitions Part

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

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

A package shall contain at most two Style Definitions parts. 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 in from the Glossary Document (§11.3.8) part.

[*Example*:

<Relationships xmlns="…">

<Relationship Id="rId3"

Type="http://…/styles" Target="styles.xml"/> </Relationships>

*end example*]

The root element for a Styles Definition part shall be styles, which is a container for one or more style elements. [*Example*: Here is the style ListBullet (which is used in a Main Document Part in §11.3.10):

<w:styles xmlns:w="…" … xml:space="preserve">

<w:style w:type="paragraph" w:styleId="ListBullet">

<w:name w:val="List Bullet"/>

<w:basedOn w:val="Text"/>

<w:autoRedefine/>

<w:rsid w:val="00081289"/>

<w:pPr>

<w:pStyle w:val="ListBullet"/>

<w:numPr>

<w:numId w:val="1"/>

</w:numPr>

<w:tabs>

<w:tab w:val="clear" w:pos="360"/>

</w:tabs>

<w:ind w:start="648"/>

</w:pPr>

</w:style> </w:styles>

*end example*]

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

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

#### 11.3.13 Web Settings Part

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

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

A package shall contain at most two Web Settings parts. 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*:

<Relationships xmlns="…">

<Relationship Id="rId3"

Type="http://…/webSettings" Target="webSettings.xml"/> </Relationships>

*end example*]

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

[*Example*:

<w:webSettings …>

<w:frameset>

…

<w:frame>

<w:sz w:val="216" />

<w:name w:val="Frame2" />

<w:sourceFileName r:id="rId1" />

</w:frame>

<w:frame>

<w:name w:val="Frame1" />

<w:sourceFileName r:id="rId2" />

</w:frame>

</w:frameset>

</w:webSettings>

*end example*]

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

 Frameset (§11.5)

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

## 11.4 Document Template

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

A *document template* can be represented by an instance of a WordprocessingML package, and contains styles, numbering definitions, and so on that are made available when documents based on that template are edited. A WordprocessingML document can refer to another document as its document template, by having a Document Settings part (§11.3.3) that contains an explicit relationship to the file location of the necessary document template using the id attribute on the attachedTemplate element.

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/attachedTemplate" Target="file:///c:\template.docx"

TargetMode="External"/>

</Relationships>

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

<w:settings … >

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

*end example*]

## 11.5 Framesets

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

A *frameset* is a WordprocessingML document which specifies the location and placement of other

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/frame" Target="frame1.docx" TargetMode="External"/>

<Relationship Id="rId2"

Type="http://…/frame" Target="frame2.docx" TargetMode="External"/> </Relationships>

The frameset document’s Web Settings part contains a frameset element that references its frames: <w:webSettings …>

<w:frameset>

…

<w:frame>

<w:sz w:val="216" />

<w:name w:val="Frame2" />

<w:sourceFileName r:id="rId1" />

</w:frame>

<w:frame>

<w:name w:val="Frame1" />

<w:sourceFileName r:id="rId2" />

</w:frame>

</w:frameset>

</w:webSettings>

*end example*]

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

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

## 11.6 Master Documents and Subdocuments

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/subDocument"

Target="Start.docx" TargetMode="External"/>

<Relationship Id="rId6"

Type="http://…/SubDocument"

Target="Middle.docx" TargetMode="External"/>

<Relationship Id="rId7"

Type="http://…/SubDocument"

Target="End.docx" TargetMode="External"/>

</Relationships>

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

<w:document xmlns:r="…" xmlns:w="…" …>

<w:body>

<w:p …>

<w:pPr>

…

</w:pPr>

</w:p>

<w:subDoc r:id="rId5"/>

…

<w:subDoc r:id="rId6"/>

…

<w:subDoc r:id="rId7"/>

…

</w:body>

</w:document>

*end example*]

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

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

## 11.7 Mail Merge Data Source

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/mailMergeSource"

Target="http://www.openxmlformats.org/data.txt"

TargetMode="External"/>

</Relationships>

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

<w:settings …>

<w:mailMerge>

…

<w:dataSource r:id="rId1" />

…

</w:mailMerge> </w:settings>

*end example*]

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

## 11.8 Mail Merge Header Data Source

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/mailMergeHeaderSource"

Target="http://www.openxmlformats.org/header.txt"

TargetMode="External"/>

</Relationships>

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

<w:settings …>

<w:mailMerge>

…

<w:headerSource r:id="rId2" />

…

</w:mailMerge> </w:settings>

*end example*]

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

## 11.9 XSL Transformation

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId8" Type="http://…/transform"

Target="http://www.openxmlformats.org/test.xsl"

TargetMode="External"/>

</Relationships>

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

<w:settings …>

…

<w:saveThroughXslt r:id="rId8" />

…

</w:settings>

*end example*]

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

# 12. SpreadsheetML

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

## 12.1 Glossary of SpreadsheetML-Specific Terms

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

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

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

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

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

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

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

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

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

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

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

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

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

**workbook** — A collection of worksheets.

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

## 12.2 Package Structure

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

* One Workbook part (12.3.23).

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

* 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 §12.3 and its subordinate clauses.

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

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

<Types xmlns="…">

<Default Extension="rels"

ContentType="application/vnd.openxmlformats-package.relationships+xml" />

<Override PartName="/workbook.xml"

ContentType="application/vnd.openxmlformats-officedocument.

spreadsheetml.sheet.main+xml" />

<Override PartName="/sheet1.xml"

ContentType="application/vnd.openxmlformats- officedocument.spreadsheetml.worksheet+xml" /> </Types>

Next, the required package-level relationship to the Workbook 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="workbook.xml" />

</Relationships>

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

<workbook xmlns="…" xmlns:r="…">

<sheets>

<sheet name="1" sheetId="1" r:id="rId1" />

</sheets>

</workbook>

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://purl.oclc.org/ooxml/officeDocument/relationships/worksheet"

Target="sheet1.xml" />

</Relationships>

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

<worksheet xmlns="…" xmlns:r="…">

<sheetData /> </worksheet>

*end example*]

[*Example*: Consider a SpreadsheetML document that contains a workbook having three worksheets. 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, Workbook (stored in the ZIP item /xl/workbook.xml), has its own relationship item:

|  |  |  |  |
| --- | --- | --- | --- |
| /\_rels/.rels |  | *Package-relationship item* | |
| /[Content\_Types].xml |  | *Content-type item* | |
| /docProps/app.xml |  | *Application-Defined File Properties part* | |
| /docProps/core.xml |  | *Core File Properties part* | |
| /xl/workbook.xml | | *Workbook part* |
| /xl/\_rels/workbook.xml.rels | | *Part-relationship item* |
| /xl/calcChain.xml | | *Calculation Chain part* |
| /xl/sharedStrings.xml | | *Shared String Table part* |
| /xl/styles.xml | | *Styles part* |
| /xl/volatileDependencies.xml | | *Volatile Dependencies part* |
| /xl/theme/theme1.xml | | *Theme part* |
| /xl/worksheets/sheet1.xml  /xl/worksheets/sheet2.xml  /xl/worksheets/sheet3.xml | | *Worksheet parts* |

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="xl/workbook.xml"/> </Relationships>

*end example*]

## 12.3 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to SpreadsheetML.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Calculation Chain | Workbook | calcChain | §12.3.1 |
| Chartsheet | Workbook | chartsheet | §12.3.2 |
| Comments | Dialogsheet, Worksheet | comments | §12.3.3 |
| Connections | Workbook | connections | §12.3.4 |
| Custom Property | Workbook | Not applicable | §12.3.5 |
| Custom XML Mappings | Workbook | MapInfo | §12.3.6 |
| Dialogsheet | Workbook | dialogsheet | §12.3.7 |
| Drawings | Chartsheet, Worksheet | wsDr | §12.3.8 |
| External Workbook References | Workbook | externalLink | §12.3.9 |
| Metadata | Workbook | metadata | §12.3.10 |
| Pivot Table | Worksheet | pivotTableDefinition | §12.3.11 |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Pivot Table Cache Definition | Pivot Table, Workbook | pivotCacheDefinition | §12.3.12 |
| Pivot Table Cache Records | Pivot Table Cache Definition | pivotCacheRecords | §12.3.13 |
| Query Table | Worksheet | queryTable | §12.3.14 |
| Shared String Table | Workbook | sst | §12.3.15 |
| Shared Workbook Revision Headers | Workbook | headers | §12.3.16 |
| Shared Workbook  Revision Log | Shared Workbook Revision Headers | revisions | §12.3.17 |
| Shared Workbook User Data | Workbook | users | §12.3.18 |
| Single Cell Table Definitions | Dialogsheet, Worksheet | singleXmlCells | §12.3.19 |
| Styles | Workbook | styleSheet | §12.3.20 |
| Table Definition | Dialogsheet, Worksheet | table | §12.3.21 |
| Volatile  Dependencies | Workbook | volTypes | §12.3.22 |
| Workbook | SpreadsheetML package | workbook | §12.3.23 |
| Worksheet | Workbook | worksheet | §12.3.24 |

*end note*]

#### 12.3.1 Calculation Chain Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId7"

Type="http://…/calcChain" Target="calcChain.xml"/> </Relationships>

*end example*]

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

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

<calcChain xmlns="…">

<c r="E8" i="1"/>

<c r="D8"/>

<c r="F8" s="1"/>

</calcChain>

*end example*]

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

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

#### 12.3.2 Chartsheet Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/chartsheet" Target="chartsheets/sheet1.xml"/> <Relationship Id="rId5"

Type="http://…/chartsheet" Target="chartsheets/sheet2.xml"/> <Relationship Id="rId6"

Type="http://…/chartsheet" Target="chartsheets/sheet3.xml"/> </Relationships>

*end example*]

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

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

<chartsheet xmlns:r="…" …>

<sheetViews>

<sheetView scale="64"/>

</sheetViews>

<drawing r:id="rId1"/> </chartsheet>

*end example*]

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

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

 Printer Settings (§15.2.15)

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

 Drawings (§12.3.8)

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

#### 12.3.3 Comments Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/comments" Target="../comments2.xml"/> </Relationships>

*end example*]

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

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

<comments xmlns:st="…" >

<authors>

<author>James Jones</author>

<author>Mary Smith</author>

</authors>

<commentList>

<comment r="C7" authorId="0">

<text>

<st:r>

<st:rPr>

…

</st:rPr>

<st:t>James Jones:</st:t>

</st:r>

<st:r>

<st:rPr>

…

</st:rPr>

<st:t>Check that this date is correct.</st:t>

</st:r>

</text>

</comment>

<comment r="E7" authorId="1">

…

</comment>

</commentList> </comments>

*end example*]

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

#### 12.3.4 Connections Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/connections" Target="connections.xml"/> </Relationships>

*end example*]

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

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

<connections …>

<connection id="1" odcFile="…" keepAlive="1" name="…" type="5" refreshedVersion="2" background="1" saveData="1">

<dbPr connection="Provider=MSDASQL.1;Persist Security Info=True;Data

Source=dBASE Files;Extended Properties=&quot;DSN=dBASE Files;DBQ=E:\MY

DOCUMENTS;DefaultDir=E:\MY

DOCUMENTS;DriverId=533;MaxBufferSize=2048;PageTimeout=5;&quot;;Initial

Catalog=E:\MY DOCUMENTS" command="`E:\MY DOCUMENTS`\`ADDRESS`" commandType="3"/>

</connection>

<connection id="2" …>

<dbPr … />

</connection>

<connection id="3" …>

<dbPr … />

</connection>

</connections>

*end example*]

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

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

#### 12.3.5 Custom Property Part

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

This part supports the storage of user-defined data.

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId7"

Type="http://…/customProperty" Target="CustomProperty.xml"/> </Relationships>

where the contents of CustomProperty.xml contain the following XML

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<CustomApplicationData xmlns="…">

<CustomProperty name="PropertyName" value="PropertyValue" /> </CustomApplicationData>

*end example*]

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

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

#### 12.3.6 Custom XML Mappings Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId9"

Type="http://…/xmlMaps" Target="xmlMaps.xml"/>

</Relationships>

*end example*]

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

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

<mapInfo SelectionNamespaces="">

<Schema ID="Schema1">

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element nillable="true" name="names">

<xsd:complexType>

<xsd:sequence minOccurs="0">

<xsd:element minOccurs="0" maxOccurs="unbounded" nillable="true" name="name" form="unqualified">

<xsd:complexType>

<xsd:sequence minOccurs="0">

<xsd:element minOccurs="0" nillable="true" type="xsd:string" name="firstname" form="unqualified"/>

<xsd:element minOccurs="0" nillable="true" type="xsd:string" name="initial" form="unqualified"/>

<xsd:element minOccurs="0" nillable="true" type="xsd:string" name="lastName" form="unqualified"/> </xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

</Schema>

<Map ID="1" Name="names\_Map" RootElement="names" SchemaID="Schema1"

ShowImportExportValidationErrors="false" AutoFit="true"

Append="false"

PreserveSortAFLayout="true" PreserveFormat="true">

<DataBinding FileBinding="Test.xml" DataBindingLoadMode="1"/>

</Map>

</mapInfo>

*end example*]

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

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

#### 12.3.7 Dialogsheet Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/dialogsheet" Target="dialogsheets/sheet1.xml"/> </Relationships>

The Workbook part contains the following:

<workbook xmlns:r="…" …>

…

<sheets>

…

<sheet name="Dialog1" sheetId="4" r:id="rId2"/>

</sheets>

…

</workbook>

*end example*]

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

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

<dialogsheet xmlns:r="…" …>

<sheetPr>

<pageSetUpPr/>

</sheetPr>

<sheetViews>

…

</sheetViews>

…

<legacyDrawing r:id="rId1"/> </dialogsheet>

*end example*]

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

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

 Printer Settings (§15.2.15)

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

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

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

#### 12.3.8 Drawings Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http:// …/drawing" Target="../drawings/drawing1.xml"/> </Relationships>

*end example*]

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

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

<xdr:wsDr xmlns:xdr="…" xmlns:a="…">

<xdr:absoluteAnchor>

<xdr:pos x="1518046" y="-1443632"/>

<xdr:extents cx="8587382" cy="5848945"/>

<xdr:graphicFrame macro="">

<xdr:nvGraphicFramePr>

<xdr:cNvPr id="24" name="Chart 24" descr=""/>

<xdr:cNvGraphicFramePr/>

</xdr:nvGraphicFramePr>

<xdr:xfrm>

<a:off x="0" y="0"/>

<a:ext cx="0" cy="0"/>

</xdr:xfrm>

<a:graphic>

<a:graphicData uri="http://…/chart">

<a:chart relId="rId1"/>

</a:graphicData>

</a:graphic>

</xdr:graphicFrame> <xdr:clientData/>

</xdr:absoluteAnchor> </xdr:wsDr>

*end example*]

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

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

* 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)

* Hyperlinks (§15.3)  Images (§15.2.14)

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

#### 12.3.9 External Workbook References Part

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

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

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

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

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

<workbook xmlns:r="…"/>

…

<sheets>

<sheet name="Sheet1" sheetId="1" r:id="rId1"/>

<sheet name="Sheet2" sheetId="2" r:id="rId2"/>

<sheet name="Sheet3" sheetId="3" r:id="rId3"/>

</sheets>

…

<externalReferences>

<externalReference r:id="rId4"/>

</externalReferences>

…

</workbook>

That part's relationship item contains the following:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/externalLink"

Target="externalReferences/externalReference1.xml"/> </Relationships>

*end example*]

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

[*Example*: externalReference1.xml contains:

<externalLink xmlns:r="…" … r:id="rId1">

<externalBook>

<sheetNames>

<sheetName val="Sheet1"/>

<sheetName val="Sheet2"/>

<sheetName val="Sheet3"/>

</sheetNames>

<sheetDataSet>

<sheetData sheetId="0">

<row r="7">

<cell r="C8">

<v>0</v>

</cell>

</row>

</sheetData>

<sheetData sheetId="1"/>

<sheetData sheetId="2"/>

</sheetDataSet>

</externalBook>

</externalLink>

This part's relationship item contains the following:

<Relationships …>

<Relationship Id="rId1"

Type="…/externalReference"

Target="wb2.xlsx" TargetMode="External"/> </Relationships>

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

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

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

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

#### 12.3.10 Metadata Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId10"

Type="http://…/sheetMetadata" Target="metadata.xml"/> </Relationships>

*end example*]

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

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

<metadata …>

<metadataTypes count="1">

<metadataType name="XLMDX" minSupportedVersion="120000" copy="1" pasteAll="1" pasteValues="1" merge="1" splitFirst="1" rowColShift="1" clearFormats="1" clearComments="1" assign="1" coerce="1"/>

</metadataTypes>

<metadataStrings count="2">

<s v="externalData"/>

<s v="[Account].[All Account]"/>

</metadataStrings>

<mdxMetadata count="1">

<m n="0" f="m">

<t c="1">

<n v="1"/>

</t>

</m>

</mdxMetadata>

<valueMetadata count="1">

<b>

<r t="1" v="0"/>

</b>

</valueMetadata>

</metadata>

The corresponding Connections part contains the following:

<connections …>

<connection id="1" odcFile="…" keepAlive="1" name="externalData" description="…" type="5" refreshedVersion="3" background="1">

<dbPr connection="Provider=MSOLAP.2;…" command="Budget" commandType="1"/>

<olapPr sendLocale="1" rowDrillCount="1000" serverFill="1" serverNumberFormat="1" serverFont="1" serverFontColor="1"/>

</connection>

</connections>

The corresponding Volatile Dependencies part contains the following:

<volTypes …">

<volType type="cubeFunctions">

<main first="externalData">

<tp t="e">

<v>#N/A</v>

<stp>1</stp>

<r r="B3" s="1"/>

</tp>

</main>

</volType>

</volTypes>

The corresponding Pivot Table Cache Definition part contains the following:

<pivotCacheDefinition … saveData="0" refreshedBy="…" refreshedDateIso="2005-11-28T16:55:44" backgroundQuery="1" createdVersion="3" refreshedVersion="3" recordCount="0">

<cacheSource type="external" connectionID="1"/>

<cacheFields count="0"/>

<cacheHierarchies count="6">

…

</cacheHierarchies>

<kpis count="0"/>

<tupleCache>

<queryCache count="3">

<query mdx="[product].[category]"/>

<query mdx=""/>

<query mdx="[Account].[All Account]"> <tpls c="1">

<tpl hier="0" item="4294967295"/>

</tpls>

</query>

</queryCache>

</tupleCache>

</pivotCacheDefinition>

*end example*]

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

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

#### 12.3.11 Pivot Table Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/pivotTable" Target="../pivotTables/pivotTable1.xml"/> <Relationship Id="rId2"

Type="http://…/pivotTable" Target="../pivotTables/pivotTable2.xml"/> </Relationships>

*end example*]

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

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

<pivotTableDefinition … cache="4" applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="0" applyPatternFormats="0" applyAlignmentFormats="0" applyWidthHeightFormats="1" dataCaption="Data" updatedVersion="3" minRefreshableVersion="3" useAutoFormatting="1" itemPrintTitles="1" createdVersion="3" indent="0" outline="1" outlineData="1">

<location ref="H4:H5" firstHeaderRow="1" firstDataRow="1" firstDataCol="0"/>

<pivotFields count="1">

<pivotField dataField="1" numFmtId="0" outline="1" subtotalTop="1" showAll="0" measureFilter="0" sortType="manual"/>

</pivotFields>

<rowItems count="1">

<i t="data"/>

</rowItems>

<colItems count="1">

<i t="data"/>

</colItems>

<dataFields count="1">

<dataField name="Sum of 1000" fld="0" subtotal="average" baseField="0" baseItem="0" numFmtId="0"/>

</dataFields>

<tableStyle name="TableStyle2" showRowHeaders="1" showColHeaders="1" showRowStripes="1" showColStripes="1"/> </pivotTableDefinition>

*end example*]

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

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

 Pivot Table Cache Definition (§12.3.12).

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

#### 12.3.12 Pivot Table Cache Definition Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type=http://…/pivotCacheDefinition

Target="../pivotCache/pivotCacheDefinition2.xml"/> </Relationships>

*end example*]

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

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

<pivotCacheDefinition … r:id="rId1" refreshedBy="John Jones" refreshedDateIso="2005-11-18T16:47:49" createdVersion="3" refreshedVersion="3" recordCount="11">

<cacheSource type="worksheet">

<worksheet range="C4:C15" sheet="Sheet1"/>

</cacheSource>

<cacheFields count="1">

<cacheField name="1000">

<sharedItems containsSemiMixedTypes="0" containsString="0" containsNumber="1" containsInteger="1" minValue="234 maxValue="2543"/>

</cacheField>

</cacheFields>

</pivotCacheDefinition>

*end example*]

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

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

 Pivot Table Cache Records (§12.3.13).

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

#### 12.3.13 Pivot Table Cache Records Part

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

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

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

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/pivotCacheRecords" Target="pivotCacheRecords2.xml"/> </Relationships>

*end example*]

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

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

<pivotCacheRecords … count="11">

<r>

<n v="1234"/>

</r>

…

<r>

<n v="876"/>

</r>

</pivotCacheRecords>

*end example*]

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

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

#### 12.3.14 Query Table Part

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

An instance of this part type contains information that describes how the source table is connected to an external data source, and defines the properties that is used when this table is refreshed from that source.

A package is permitted to contain one Query Table part per table, and each of those parts shall be the target of an implicit relationship from the corresponding Table Definitions (§12.3.21) part.

[*Example*: The following Table part-relationship item contains a relationship to the Query Table part corresponding to the connections details for that table. These parts are stored in the ZIP items ../queryTables/queryTable*n*.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/queryTable"

Target="../queryTables/queryTable1.xml"/> </Relationships>

*end example*]

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

[*Example*: queryTable2.xml deals with a connection to a database file having the seven fields shown:

<queryTable … name="+Connect to New Data Source\_1" growShrinkType="insertDelete" connectionId="2" autoFormatId="16" applyNumberFormats="0" applyBorderFormats="0" applyFontFormats="1" applyPatternFormats="1" applyAlignmentFormats="0" applyWidthHeightFormats="0"> <queryTableRefresh nextId="8">

<queryTableFields count="7">

<queryTableField id="1" name="ACCOUNT"/>

<queryTableField id="2" name="CHECKNUM"/>

<queryTableField id="3" name="DATE"/>

<queryTableField id="4" name="AMOUNT"/>

<queryTableField id="5" name="PAYEE"/>

<queryTableField id="6" name="CHARGECODE"/>

<queryTableField id="7" name="DESCRIPT"/>

</queryTableFields>

</queryTableRefresh> </queryTable>

*end example*]

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

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

#### 12.3.15 Shared String Table Part

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

An instance of this part type contains one occurrence of each unique string that occurs on all worksheets in a workbook.

A package shall contain exactly one Shared String Table part, and that part shall be the target of an implicit relationship from the Workbook part (§12.3.23).

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared String Table part, which is stored in the ZIP item sharedStrings.xml:

<Relationships xmlns="…">

<Relationship Id="rId6"

Type="http://…/sharedStrings" Target="sharedStrings.xml"/> </Relationships>

*end example*]

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

[*Example*: Here are three of the six strings used in the worksheets:

<sst xmlns:st="…" … totalCount="6" uniqueCount="6">

<sstItem>

<t>Expenses Log</t>

</sstItem>

<sstItem>

<t>Period Start</t>

</sstItem>

<sstItem>

<t>Period End</t>

</sstItem>

…

</sst>

*end example*]

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

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

#### 12.3.16 Shared Workbook Revision Headers Part

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

An instance of this part type contains information about each of the editing sessions performed on the parent workbook at the worksheet level (worksheets added and rearranged in each session).

A package shall contain at most one Shared Workbook Revision Headers part. If it exists, that part shall be the target of an implicit relationship from the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared Workbook Revision Headers part, which is stored in the ZIP item handout revisions/revisionHeaders.xml:

<Relationships xmlns="…">

<Relationship Id="rId9"

Type="http://…/revisionHeaders"

Target="revisions/revisionHeaders.xml"/> </Relationships>

*end example*]

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

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

<headers xmlns:r="…" guid="{233BEE23-EB5C-4542-905D-0230EFFED88B}" diskRevisions="1" revisionId="4" version="3">

<header guid="…" dateTime="…" maxSheetId="4" userName="…" r:id="rId1">

<sheetIdMap count="3">

…

</sheetIdMap>

</header>

…

<header guid="…" dateTime="…" maxSheetId="4" userName="…" r:id="rId3">

<sheetIdMap count="3">

…

</sheetIdMap>

</header> </headers>

*end example*]

A Shared Workbook Revision Headers part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be Internal).

A Shared Workbook Revision Headers part is permitted to have explicit relationships to the following parts defined by ISO/IEC 29500:

 Shared Workbook Revision Log (§12.3.17)

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

#### 12.3.17 Shared Workbook Revision Log Part

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

An instance of this part type contains information about edits performed on individual cells in the parent workbook’s worksheets in each editing session.

A package shall contain one Shared Workbook Revision Log part for each session's set of changes, and those parts shall be the target of an explicit relationship from the Shared Workbook Revision Headers (§12.3.16) part.

[*Example*: The following Shared Workbook Revision Headers part-relationship item contains a number of relationships to Shared Workbook Revision Log parts, which are stored in the ZIP item revisionLog*N*.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/revisionLog" Target="revisionLog1.xml"/>

…

<Relationship Id="rId6"

Type="http://…/revisionLog" Target="revisionLog6.xml"/> </Relationships>

*end example*]

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

[*Example*: revisionLog2.xml contains the following:

<revisions xmlns:xs="…" …>

<rfmt sheetId="1" sqref="B4:B15">

<dxf>

<xs:fill>

<xs:pattern patternType="solid">

<xs:fgColor type="icv" val="64"/>

<xs:bgColor type="rgb" val="4278252287"/>

</xs:pattern>

</xs:fill>

</dxf>

</rfmt>

<rcv guid="{CBCE5672-5A4D-48C9-A120-F72804F8CF64}" action="delete"/>

<rcv guid="{CBCE5672-5A4D-48C9-A120-F72804F8CF64}" action="add"/> </revisions>

*end example*]

A Shared Workbook Revision Log part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be Internal).

A Shared Workbook Revision Log part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

#### 12.3.18 Shared Workbook User Data Part

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

An instance of this part type contains a list of all the users that are sharing the parent workbook.

A package shall contain at most one Shared Workbook User Data part, and that part shall be the target of an implicit relationship in the Workbook (§12.3.23) part.

[*Example*: The following Workbook part-relationship item contains a relationship to the Shared Workbook User Data part, which is stored in the ZIP item revisions/userNames.xml:

<Relationships xmlns="…">

Relationship Id="rId8"

Type="http://…/usernames" Target="revisions/userNames.xml"/> </Relationships>

*end example*]

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

[*Example*: userNames.xml shows that there are two users sharing this workbook:

<users … count="2">

<usrinfo guid="{B5A024F7-40BE-4A48-9B6D-B1655241C84D}"

name="Mary Jones" id="-264292310" dateTime="2005-11-18T18:53:16"/>

<usrinfo …/> </users>

*end example*]

A Shared Workbook User Data part shall be located within the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be Internal).

A Shared Workbook User Data part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

#### 12.3.19 Single Cell Table Definitions Part

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

An instance of this part type contains information on how to map non-repeating elements from a custom XML file into cells in a worksheet. [*Note*: Repeating custom XML elements are mapped using a Table (§12.3.21). *end note*]

A package shall contain at most one Single Cell Table Definitions part per worksheet, and that part shall be the target of an implicit relationship from a Worksheet (§12.3.24) part. A Single Cell Table Definitions part can describe one or more single cell table definitions for any given worksheet.

[*Example*: The following Worksheet part-relationship item contains a relationship to the Single Cell Table Definitions part, which is stored in the ZIP item ../tables/tableSingleCells1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/tableSingleCells"

Target="../tables/tableSingleCells1.xml"/> </Relationships>

*end example*]

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

[*Example*: A worksheet contains two single cell table definitions; e.g., ../tables/tableSingleCells1.xml contains the following, where the elements id and count are nested inside element names:

<singleXmlCells …>

<singleCell id="1" name="Table1" displayName="Table1" ref="B4">

<cellPr id="1" uniqueName="id">

<xmlPr mapId="1" xpath="/names/id" xmlDataType="string"/>

</cellPr>

</singleCell>

<singleCell id="2" name="Table2" displayName="Table2" ref="B7">

<cellPr id="1" uniqueName="count">

<xmlPr mapId="1" xpath="/names/count" xmlDataType="integer"/>

</cellPr>

</singleCell>

</singleXmlCells>

*end example*]

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

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

#### 12.3.20 Styles Part

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

An instance of this part type contains all the characteristics for all the cells in the workbook. Such information includes numeric and text formatting, alignment, font, color, and border.

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

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

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/styles" Target="styles.xml"/>

</Relationships>

*end example*]

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

[*Example*:

<styleSheet xmlns="…">

<numFmts count="5">

<numFmt numFmtId="164" formatCode="&quot;$&quot;#,##0.00"/>

<numFmt numFmtId="165" formatCode="&quot;Yes&quot;;&quot;Yes&quot;;&quot;No&quot;"/>

<numFmt numFmtId="166" formatCode="&quot;True&quot;;&quot;True&quot;;&quot;False&quot;"/>

<numFmt numFmtId="167" formatCode="&quot;On&quot;;&quot;On&quot;;&quot;Off&quot;"/>

<numFmt numFmtId="168" formatCode="[$€-2]\ #,##0.00\_);[Red]\([$€-2]\ #,##0.00\)"/> </numFmts> <fonts count="5">

…

</fonts>

<fills count="4">

…

</fills>

<borders count="1">

…

</borders>

…

<colors>

…

</colors>

</styleSheet>

*end example*]

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

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

#### 12.3.21 Table Definition Part

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

An instance of this part type contains a description of a single table and its autofilter information. (The data for the table is stored in the corresponding Worksheet part.)

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

[*Example*: The following Worksheet part-relationship item contains relationships to two Table Definition parts, which are stored in the ZIP items ../tables/table*N*.xml:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/table" Target="../tables/table1.xml"/> <Relationship Id="rId3"

Type="http://…/table" Target="../tables/table2.xml"/> </Relationships>

*end example*]

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

[*Example*: table2.xml describes a table that spans a 2-column range of cells, F2:G19:

<table xmlns:af="…" … id="2" name="Table2" displayName="Table2" ref="F2:G19" totalsRowShown="0" headerRowDxfId="7">

<autoFilter ref="F2:G19"/>

<tableColumns count="2">

<tableColumn id="1" name="Salesman" dataDxfId="9" totalsRowDxfId="6"/>

<tableColumn id="2" name="Units" dataDxfId="8" totalsRowDxfId="5"/>

</tableColumns>

<tableStyle name="TableStyle2" showFirstColumn="0" showLastColumn="0" showRowStripes="1" showColumnStripes="1"/> </table>

When the filter "Salesman equal to Smith" is applied, the autoFilter element in table2.xml is extended, as follows:

<autoFilter ref="F2:G19">

<af:filterColumn colId="0">

<af:filters>

<af:filter val="Smith"/>

</af:filters>

</af:filterColumn> </autoFilter>

*end example*]

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

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

 Query Table (§12.3.14)

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

#### 12.3.22 Volatile Dependencies Part

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

An instance of this part type contains information involving real-time data formulas in a workbook. Real-time data formulas return values that change over time — often every few seconds — and require connectivity to programs outside of the workbook to retrieve their data. In cases where those programs are not available, realtime data formulas can use the information stored in the Volatile Dependencies part to calculate results, rather than generate errors. More information on real-time data functions can be found in §18.17.7.284 and §18.17.7.65 through §18.17.7.71.

A package shall contain exactly one Volatile Dependencies part, and that part shall be the target of an implicit relationship from the Workbook (§12.3.23) part.

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

<Relationships xmlns="…">

<Relationship Id="rId8"

Type="http://…/volatileDependencies"

Target="volatileDependencies.xml"/> </Relationships>

*end example*]

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

[*Example*:

<volTypes xmlns="…"/>

*end example*]

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

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

#### 12.3.23 Workbook Part

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

An instance of this part type contains workbook data and references to all of its worksheets.

A package shall contain exactly one Workbook part, and that part shall be the target of a relationship in the package-relationship item.

[*Example*: The following SpreadsheetML package-relationship item contains a relationship to the Workbook part, which is stored in the ZIP item xl/workbook.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/officeDocument" Target="xl/workbook.xml"/> </Relationships>

*end example*]

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

[*Example*: This workbook has three worksheets, named January, February, and March:

<workbook xmlns="…" xmlns:r="…">

<fileVersion lastEdited="4" lowestEdited="4" rupBuild="3417"/>

<bookViews>

<workbookView xWindow="240" yWindow="15" windowWidth="8505" windowHeight="6240"/>

</bookViews>

<sheets>

<sheet name="January" sheetId="1" r:id="rId1"/>

<sheet name="February" sheetId="2" r:id="rId2"/>

<sheet name="March" sheetId="3" r:id="rId3"/>

</sheets>

<workbookPr showObjects="all"/>

<webPublishing codePage="1252"/>

<calcPr calcId="122211" fullCalcOnLoad="1"/> </workbook>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Calculation Chain (§12.3.1)
* Cell Metadata (§12.3.10)
* Connections (§12.3.4)
* Custom XML Mappings (§12.3.6)
* Custom XML Data Storage (§15.2.4)
* Shared String Table (§12.3.15)
* Shared Workbook Revision Headers (§12.3.16)
* Shared Workbook User Data (§12.3.18)
* Styles (§12.3.20)
* Theme (§14.2.7)
* Thumbnail (§15.2.16)
* Volatile Dependencies (§12.3.22)

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

* Chartsheet (§12.3.2)
* Dialogsheet (§12.3.7)
* External Workbook References (§12.3.8)
* Pivot Table Cache Definition (§12.3.12)
* Worksheet (§12.3.24)

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

#### 12.3.24 Worksheet Part

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

An instance of this part type contains all the data, formulas, and characteristics associated with a given worksheet.

A package shall contain exactly one Worksheet part per worksheet, and those parts shall be the target of an explicit relationship from the Workbook (§12.3.23) part. Specifically, the id attribute on the sheet element shall reference the desired worksheet part.

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

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/worksheet" Target="worksheets/sheet1.xml"/> <Relationship Id="rId2"

Type="http://…/worksheet" Target="worksheets/sheet2.xml"/> <Relationship Id="rId3"

Type="http://…/worksheet" Target="worksheets/sheet3.xml"/> </Relationships>

*end example*]

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

[*Example*: This worksheet, has cells in the range B1 to F8. Row 8 contains three cells whose values are calculated using the following formulas: D8=SUM(D5:D7), E8=SUM(E5:E7), and F8= D8-E8:

<worksheet xmlns="…" …>

<sheetPr/>

<dimension range="B1:F8"/>

…

<sheetData>

<row r="1" spans="2:6" ht="360">

<c r="B1" s="1" t="s">

<v>0</v>

</c>

</row>

…

<row r="8" spans="2:6" ht="360">

<c r="D8" s="5">

<f>SUM(D5:D7)</f>

<v>2280.5299999999997</v>

</c>

<c r="E8" s="5">

<f>SUM(E5:E7)</f>

<v>1251.31</v>

</c>

<c r="F8" s="6">

<f>D8-E8</f>

<v>1029.2199999999998</v>

</c>

</row>

</sheetData>

…

</worksheet>

*end example*]

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

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

* Comments (§12.3.3)
* Pivot Table Definitions (§12.3.11)
* Printer Settings (§15.2.15)
* Query Table Part (§12.3.14)
* Single Cell Table Definitions (§12.3.19)
* Table Definition (§12.3.21)

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

* Drawings (§12.3.8)
* Embedded Control Persistence (§15.2.9)
* Embedded Object (§15.2.10)
* Embedded Package (§15.2.11)
* Hyperlinks (§15.3)  Images (§15.2.14)

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

## 12.4 External Workbooks

|  |  |
| --- | --- |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/externalLinkPath |

An *external workbook* is a SpreadsheetML package whose contents are being referenced by the current SpreadsheetML package. When a package refers to external workbooks, it shall store the location of those workbooks using this relationship.

A package is permitted to contain one or more External Workbook relationships, and those relationships shall be an explicit relationship from the External Workbook References (§12.3.9) part.

[*Example*: An External Workbook References part, which references the package c:\sourceData.xlsx would have an External Workbook relationship, which points at that file:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/externalLinkPath"

Target="c:\sourceData.xlsx" TargetMode="External"/> </Relationships>

*end example*]

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

# 13. PresentationML

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

## 13.1 Glossary of PresentationML-Specific Terms

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

**handout** — A printed set of slides that can be handed out to an audience for future reference. **note** — A slide annotation, reminder, or piece of text intended for the presenter or the audience. **presentation** — A collection of slides intended to be viewed by an audience. **slide** — A frame containing one or more pieces of text and/or images.

**slide layout** — The organization of elements on a slide.

## 13.2 Package Structure

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

* One Presentation part (§13.3.6).

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

* 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 §13.3 and its subordinate clauses.

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

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

<Types xmlns="…">

<Default Extension="rels"

ContentType="application/vnd.openxmlformats- package.relationships+xml"/>

<Override PartName="/presentation.xml" ContentType="application/vnd.openxmlformats-

officedocument.presentationml.presentation.main+xml"/> </Types>

Next, the single required relationship (the package-level relationship to the Main Presentation 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="presentation.xml"/>

</Relationships>

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

<p:presentation xmlns:p="…">

<p:notesSz cx="913607" cy="913607"/> </p:presentation>

*end example*]

[*Example*: Consider a simple PresentationML document containing two slides, which both use an image as a template. 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, a number of parts have their own relationship items:

|  |  |  |
| --- | --- | --- |
| /\_rels/.rels |  | *Package-relationship item* |
| /[Content\_Types].xml |  | *Content-type item* |
| /docProps/app.xml *part* |  | *Application-Defined File Properties* |
| /docProps/core.xml |  | *Core File Properties part* |
| /docProps/custom.xml |  | *Custom File Properties part* |
| /docProps/thumbnail.wmf |  | *Package thumbnail image* |
| /ppt/presentation.xml |  | *Presentation part* |
| /ppt/\_rels/presentation.xml.rels |  | *Part-relationship item* |
| /ppt/presProps.xml |  | *Presentation Properties part* |
| /ppt/tableStyles.xml |  | *Table Styles part* |
| /ppt/viewProps.xml |  | *View Properties part* |

/ppt/handoutMasters/handoutMaster1.xml *Handout Master part*

/ppt/handoutMasters/\_rels/handoutMaster1.xml.rels

*Part-relationship item*

/ppt/media/image1.jpeg *Slide template image*

/ppt/notesMasters/notesMaster1.xml *Notes Master part*

/ppt/notesMasters/\_rels/notesMaster1.xml.rels

*Part-relationship item*

/ppt/notesSlides/notesSlide1.xml *Notes Slide parts*

/ppt/notesSlides/notesSlide2.xml

/ppt/notesSlides/\_rels/notesSlide1.xml.rels

*Part-relationship items*

/ppt/notesSlides/\_rels/notesSlide2.xml.rels

/ppt/slideLayouts/slideLayout1.xml *Slide Layout parts 1–6*

/ppt/slideLayouts/slideLayout2.xml

/ppt/slideLayouts/slideLayout3.xml

/ppt/slideLayouts/slideLayout4.xml

/ppt/slideLayouts/slideLayout5.xml

/ppt/slideLayouts/slideLayout6.xml

/ppt/slideLayouts/\_rels/slideLayout1.xml.rels

*Part-relationship items*

/ppt/slideLayouts/\_rels/slideLayout2.xml.rels

/ppt/slideLayouts/\_rels/slideLayout3.xml.rels

/ppt/slideLayouts/\_rels/slideLayout4.xml.rels

/ppt/slideLayouts/\_rels/slideLayout5.xml.rels

/ppt/slideLayouts/\_rels/slideLayout6.xml.rels

/ppt/slideMasters/slideMaster1.xml *Slide Master part*

/ppt/slideMasters/\_rels/slideMaster1.xml.rels

|  |  |  |
| --- | --- | --- |
|  |  | *Part-relationship item* |
| /ppt/slides/slide1.xml  /ppt/slides/slide2.xml |  | *Slide parts* |
| /ppt/slides/\_rels/slide1.xml.rels  /ppt/slides/\_rels/slide2.xml.rels |  | *Part-relationship items* |
| /ppt/theme/theme1.xml  /ppt/theme/theme2.xml  /ppt/theme/theme3.xml |  | *Theme parts* |

/ppt/theme/themeOverride1.xml *Theme Override parts*

/ppt/theme/themeOverride2.xml

/ppt/theme/themeOverride3.xml

/ppt/theme/themeOverride4.xml

/ppt/theme/themeOverride5.xml

/ppt/theme/themeOverride6.xml

/ppt/theme/themeOverride7.xml

/ppt/theme/themeOverride8.xml

/ppt/theme/themeOverride9.xml

/ppt/theme/themeOverride10.xml

The package-relationship item contains the following:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/officeDocument" Target="ppt/presentation.xml"/>

<Relationship Id="rId3"

Type="http://…/core-properties" Target="docProps/core.xml"/>

<Relationship Id="rId2"

Type="http://…/thumbnail" Target="docProps/thumbnail.wmf"/>

<Relationship Id="rId4"

Type="http://…/extended-properties" Target="docProps/app.xml"/> </Relationships>

*end example*]

## 13.3 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to PresentationML.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Comment Authors | Presentation | cmAuthorLst | §13.3.1 |
| Comments | Slide | cmLst | §13.3.2 |
| Handout Master | Presentation | handoutMaster | §13.3.3 |
| Notes Master | Notes Slide, Presentation | notesMaster | §13.3.4 |
| Notes Slide | Slide | notes | §13.3.5 |
| Presentation | PresentationML package | presentation | §13.3.6 |
| Presentation Properties | Presentation | presentationPr | §13.3.7 |
| Slide | Presentation | sld | §13.3.8 |
| Slide Layout | Slide Master, Notes Slide, | sldLayout | §13.3.9 |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
|  | Presentation, Slide, Slide Master |  |  |
| Slide Master | Presentation, Slide Layout | sldMaster | §13.3.10 |
| Slide Synchronization Data | Slide | sldSyncPr | §13.3.11 |
| User-Defined Tags | Presentation, Slide | tagLst | §13.3.12 |
| View Properties | Presentation | viewPr | §13.3.13 |

*end note*]

#### 13.3.1 Comment Authors Part

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

An instance of this part type contains information about each author who has added a comment to the document. That information includes the author's name, initials, a unique author-ID, a last-comment-index-used count, and a display color. (The color can be used when displaying comments to distinguish comments from different authors.)

A package shall contain at most one Comment Authors part. If it exists, that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part relationship item contains a relationship to the Comment Authors part, which is stored in the ZIP item commentAuthors.xml:

<Relationships xmlns="…">

<Relationship Id="rId8"

Type="http://…/commentAuthors" Target="commentAuthors.xml"/> </Relationships>

*end example*]

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

[*Example*: Two people have authored comments in this document: Mary Smith and Peter Jones. Her initials are "mas", her author-ID is 0, and her comments' display color index is 0. Since Mary's last-comment-index-used value is 3, the next comment-index to be used for her is 4. His initials are "pjj", his author-ID is 1, and his comments' display color index is 1. Since Peter's last-comment-index-used value is 1, the next comment-index to be used for him is 2:

<p:cmAuthorLst xmlns:p="…" …>

<p:cmAuthor id="0" name="Mary Smith" initials="mas" lastIdx="3" clrIdx="0"/>

<p:cmAuthor id="1" name="Peter Jones" initials="pjj" lastIdx="1" clrIdx="1"/> </p:cmAuthorLst>

*end example*]

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

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

#### 13.3.2 Comments Part

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

An instance of this part type contains the comments for a single slide. Each comment is tied to its author via an author-ID. Each comment's index number and author-ID combination are unique.

A package shall contain one Comments part for each slide containing one or more comments, and each of those parts shall be the target of an implicit relationship from its corresponding Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to a Comments part, which is stored in the ZIP item ../comments/comment2.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/comments"

Target="../comments/comment2.xml"/> </Relationships>

*end example*]

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

[*Example*: The Comments part contains three comments, two created by one author, and one created by another, all at the dates and times shown. The index numbers are assigned on a per-author basis, starting at 1 for an author's first comment:

<p:cmLst xmlns:p="…" …>

<p:cm authorId="0" dt="2005-11-13T17:00:22.071" idx="1">

<p:pos x="4486" y="1342"/>

<p:text>Comment text goes here.</p:text>

</p:cm>

<p:cm authorId="0" dt="2005-11-13T17:00:34.849" idx="2">

<p:pos x="3607" y="1867"/>

<p:text>Another comment's text goes here.</p:text> </p:cm>

<p:cm authorId="1" dt="2005-11-15T00:06:46.919" idx="1">

<p:pos x="1493" y="2927"/>

<p:text>comment …</p:text>

</p:cm>

</p:cmLst>

*end example*]

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

#### 13.3.3 Handout Master Part

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

An instance of this part type contains the look, position, and size of the slides, notes, header and footer text, date, or page number on the presentation's handout.

A package shall contain at most one Handout Master part, and it shall be the target of an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the Handout Master part, which is stored in the ZIP item handoutMasters/handoutMaster1.xml:

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/handoutMaster"

Target="handoutMasters/handoutMaster1.xml"/>

</Relationships>

*end example*]

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

[*Example*:

<p:handoutMaster xmlns:p="…">

<p:cSld name="">

…

</p:cSld>

<p:clrMap … />

</p:handoutMaster>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Theme (§14.2.7)
* Thumbnail (§15.2.16)

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

* Audio (§15.2.2)
* 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)
* Hyperlink (§15.3)
* Image (§15.2.14)
* Video (§15.2.15)

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

#### 13.3.4 Notes Master Part

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

An instance of this part type contains information about the content and formatting of all notes pages.

A package shall contain at most one Notes Master part, and that part shall be the target of an implicit relationship from the Notes Slide (§13.3.5) part, as well as an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the Notes Master part, which is stored in the ZIP item notesMasters/notesMaster1.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/notesMaster" Target="notesMasters/notesMaster1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:notesMaster xmlns:p="…">

<p:cSld name="">

…

</p:cSld>

<p:clrMap … />

</p:notesMaster>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Theme (§14.2.7)
* Thumbnail (§15.2.16)

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

* Audio (§15.2.2)
* 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)  Hyperlink (§15.3).
* Image (§15.2.14)
* Video (§15.2.15)

The Notes Master part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

#### 13.3.5 Notes Slide Part

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

An instance of this part type contains the notes for a single slide.

A package shall contain one Notes Slide part for each slide that contains notes. If they exist, those parts shall each be the target of an implicit relationship from the Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to a Notes Slide part, which is stored in the ZIP item ../notesSlides/notesSlide1.xml:

<Relationships xmlns="…">

<Relationship Id="rId3"

Type="http://…/notesSlide" Target="../notesSlides/notesSlide1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:notes xmlns:p="…">

<p:cSld name="">

…

</p:cSld>

<p:clrMapOvr>

<a:masterClrMapping/>

</p:clrMapOvr> </p:notes>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Notes Master (§13.3.4)
* Theme Override(§14.2.8)  Thumbnail (§15.2.16)

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

* Audio (§15.2.2)
* 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)  Hyperlink (§15.3).
* Image (§15.2.14)  Video (§15.2.15)

The Notes Slide part shall not have implicit or explicit relationships to any other part defined by ISO/IEC 29500.

#### 13.3.6 Presentation Part

|  |  |
| --- | --- |
| Content Type: | application/vnd.openxmlformats-officedocument.presentationml.presentation.main+xml application/vnd.openxmlformats-officedocument.presentationml.slideshow.main+xml |
|  | application/vnd.openxmlformats-officedocument.presentationml.template.main+xml |
| Root  Namespace: | http://purl.oclc.org/ooxml/presentationml/main |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/officeDocument |

An instance of this part type contains the definition for a slide presentation.

A package shall contain exactly one Presentation part, and that part shall be the target of a relationship in the package-relationship item.

[*Example*: The following PresentationML's package-relationship item contains a relationship to the Presentation part, which is stored in the ZIP item ppt/presentation.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/officeDocument" Target="ppt/presentation.xml"/> </Relationships>

*end example*]

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

[*Example*: This presentation contains two slides:

<p:presentation xmlns:p="…" … >

<p:sldMasterIdLst> <p:sldMasterId xmlns:rel="http://…/relationships" rel:id="rId1"/>

</p:sldMasterIdLst>

<p:notesMasterIdLst> <p:notesMasterId

xmlns:rel="http://…/relationships" rel:id="rId4"/>

</p:notesMasterIdLst>

<p:handoutMasterIdLst> <p:handoutMasterId xmlns:rel="http://…/relationships" rel:id="rId5"/>

</p:handoutMasterIdLst>

<p:sldIdLst> <p:sldId id="267"

xmlns:rel="http://…/relationships" rel:id="rId2"/>

<p:sldId id="256" xmlns:rel="http://…/relationships" rel:id="rId3"/> </p:sldIdLst>

<p:sldSz cx="9144000" cy="6858000"/>

<p:notesSz cx="6858000" cy="9144000"/> </p:presentation>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Comment Authors (§13.3.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Font (§15.2.13)
* Presentation Properties (§13.3.7)
* Table Styles (§14.2.9)  Theme (§14.2.7)
* Thumbnail (§15.2.16)
* View Properties (§13.3.13).

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

* Notes Master (§13.3.4)
* Handout Master (§13.3.3)
* Slide (§13.3.8)
* Slide Master (§13.3.10)
* User Defined Tags (§13.3.12)

#### 13.3.7 Presentation Properties Part

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

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

A package shall contain exactly one Presentation Properties part, and that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the Presentation Properties part, which is stored in the ZIP item presProps.xml:

<Relationships xmlns="…">

<Relationship Id="rId6"

Type="http://…/presProps" Target="presProps.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:presentationPr xmlns:p="…" …>

<p:clrMru>

…

</p:clrMru>

…

</p:presentationPr>

*end example*]

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

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

#### 13.3.8 Slide Part

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

A Slide part contains the contents of a single slide.

A package shall contain one Slide part per slide, and each of those parts shall be the target of an explicit relationship from the Presentation (§13.3.6) part.

[*Example*: Consider a PresentationML document having two slides. The corresponding Presentation partrelationship item contains two relationships to Slide parts, which are stored in the ZIP items slides/slide1.xml and slides/slide2.xml:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/slide" Target="slides/slide1.xml"/> <Relationship Id="rId3"

Type="http://…/slide" Target="slides/slide2.xml"/> </Relationships>

*end example*]

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

[*Example*: slides/slide1.xml contains:

<p:sld xmlns:p="…">

<p:cSld name="">

…

</p:cSld>

<p:clrMapOvr>

…

</p:clrMapOvr>

<p:timing>

<p:tnLst>

<p:par>

<p:cTn id="1" dur="indefinite" restart="never" nodeType="tmRoot"/>

</p:par>

</p:tnLst>

</p:timing> </p:sld>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Comments (§13.3.2)
* Custom XML Data Storage (§15.2.4)
* Notes Slide (§13.3.5)
* Theme Override (§14.2.8)
* Thumbnail (§15.2.16)
* Slide Layout (§13.3.9)
* Slide Synchronization Data (§13.3.11)

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

* Audio (§15.2.2)
* 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)  Hyperlink (§15.3).
* Image (§15.2.14)
* User Defined Tags (§13.3.12)
* Video (§15.2.15)

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

#### 13.3.9 Slide Layout Part

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

An instance of this part type contains the definition for a slide layout template for this presentation. This template defines the default appearance and positioning of drawing objects on this slide type when it is created.

A package shall contain one or more Slide Layout parts, and each of those parts shall be the target of an explicit relationship in the Slide Master (§13.3.10) part, as well as an implicit relationship from each of the Slide (§13.3.8) parts associated with this slide layout.

[*Example*: The following Slide Master part-relationship item contains relationships to several Slide Layout parts, which are stored in the ZIP items ../slideLayouts/slideLayout*N*.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/slideLayout"

Target="../slideLayouts/slideLayout1.xml"/>

<Relationship Id="rId2"

Type="http://…/slideLayout"

Target="../slideLayouts/slideLayout2.xml"/>

<Relationship Id="rId3"

Type="http://…/slideLayout"

Target="../slideLayouts/slideLayout3.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:sldLayout xmlns:p="…" matchingName="" type="title" preserve="1">

<p:cSld name="Title Slide">

…

</p:cSld>

<p:clrMapOvr>

<a:masterClrMapping/>

</p:clrMapOvr> <p:timing/>

</p:sldLayout> </p:sldMaster>

*end example*]

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Slide Master (§13.3.10)
* Theme Override (§14.2.8)  Thumbnail (§15.2.16)

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

* Audio (§15.2.2)
* 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)  Hyperlink (§15.3).
* Image (§15.2.14)  Video (§15.2.15)

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

#### 13.3.10 Slide Master Part

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

An instance of this part type contains the master definition of formatting, text, and objects that appear on each slide in the presentation that is derived from this slide master.

A package shall contain one or more Slide Master parts, each of which shall be the target of an explicit relationship from the Presentation (§13.3.6) part, as well as an implicit relationship from any Slide Layout (§13.3.9) part where that slide layout is defined based on this slide master. Each can optionally be the target of a relationship in a Slide Layout (§13.3.9) part as well.

[*Example*: The following Presentation part-relationship item contains a relationship to the Slide Master part, which is stored in the ZIP item slideMasters/slideMaster1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/slideMaster" Target="slideMasters/slideMaster1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:sldMaster xmlns:p="…">

<p:cSld name="">

…

</p:cSld>

<p:clrMap … /> </p:sldMaster>

*end example*]

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

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

* Additional Characteristics (§15.2.1)
* Bibliography (§15.2.3)
* Custom XML Data Storage (§15.2.4)
* Theme (§14.2.7)
* Thumbnail (§15.2.16)

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

* Audio (§15.2.2)
* 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)  Hyperlink (§15.3).
* Image (§15.2.14)
* Slide Layout (§13.3.9)  Video (§15.2.15)

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

#### 13.3.11 Slide Synchronization Data Part

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

An instance of this part type contains properties specifying the current state of a slide that is being synchronized with a version of that slide stored on a central server.

A package shall contain zero or one Slide Synchronization Data part for each slide stored in the presentation, and that part shall be the target of an implicit relationship from the corresponding Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to the Slide Synchronization Data part, which is stored in the ZIP item slideUpdateInfo/slideUpdateInfo1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1" Type="http://…/slideUpdateInfo"

Target="slideUpdateInfo/slideUpdateInfo1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:sldSyncPr xmlns:p="…" serverSldId="1" serverSldModifiedTime="2006-08-12T01:31:08" clientInsertedTime="2006-08-12T01:34:11.227" />

*end example*]

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

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

 Slide Synchronization Server Location (§13.4)

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

#### 13.3.12 User Defined Tags Part

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

An instance of this part type contains a set of user-defined properties for an object in a presentation (each property consisting of a name/value pair).

A package shall contain zero or more User Defined Tags parts, each as the target of an explicit relationship from the corresponding Presentation (§13.3.6) or Slide (§13.3.8) part.

[*Example*: The following Slide part-relationship item contains a relationship to the User Defined Tags part, which is stored in the ZIP item tags/tag1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1" Type="http://…/tag"

Target="tags/tag1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:tagLst xmlns:p="…" >

<p:tag name="testTagName" val="testTagValue" /> </p:tagLst>

*end example*]

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

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

#### 13.3.13 View Properties Part

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

An instance of this part type contains display properties for this presentation.

A package shall contain zero or one View Properties part, and if it exists, that part shall be the target of an implicit relationship from the Presentation (§13.3.6) part.

[*Example*: The following Presentation part-relationship item contains a relationship to the View Properties part, which is stored in the ZIP item viewProps.xml:

<Relationships xmlns="…">

<Relationship Id="rId7"

Type="http://…/viewProps" Target="viewProps.xml"/> </Relationships>

*end example*]

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

[*Example*:

<p:viewPr xmlns:p="…" …>

<p:normalViewPr showOutlineIcons="0">

…

</p:normalViewPr>

<p:slideViewPr>

…

</p:slideViewPr>

<p:outlineViewPr>

…

</p:outlineViewPr>

<p:notesTextViewPr>

…

</p:notesTextViewPr>

<p:gridSpacing cx="78028800" cy="78028800"/> </p:viewPr>

*end example*]

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

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

## 13.4 HTML Publish Location

|  |  |
| --- | --- |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/htmlPubSaveAs |

When a presentation specifies an external location to which an optional copy might be pushed in the HTML format, this relationship shall be used to target the location where the HTML copy of the presentation is published.

A package shall contain one HTML Publish Location relationship for each slide linked with an HTML publish location, and that relationships shall be an explicit relationship from the corresponding Presentation Properties (§13.3.7) part.

[*Example*: A Presentation Properties part, which stores an HTML Publish Location of http://www.openxmlformats.org/test.htm contains the following relationship in that part's relationship part:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/htmlPubSaveAs"

Target="http://www.openxmlformats.org/test.htm" type=”External”/> </Relationships>

*end example*]

An HTML publish location shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be External).

## 13.5 Slide Synchronization Server Location

|  |  |
| --- | --- |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/slideUpdateUrl |

When a slide is being synchronized with a copy stored on a remote server, this relationship shall be used to target the location where the server copy of the slide is stored.

A package shall contain one Slide Synchronization Server Location relationship for each slide linked with server data, and that relationships shall be an implicit relationship from the corresponding Slide Synchronization Data (§13.3.11) part.

[*Example*: A Slide Synchronization Data part that stores information about a slide that is synchronized with a server located at http://www.openxmlformats.org/slides/ contains the following relationship in that part's relationship part item:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/slideupdateUrl"

Target="http://www.openxmlformats.org/slides/" type=”External”/> </Relationships>

*end example*]

A slide synchronization server location shall be located external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element shall be External).

# 14. DrawingML

The relationship items and parts defined in this clause are used by one or more of WordprocessingML (§11), SpreadsheetML (§12), and PresentationML (§13) environments.

## 14.1 Glossary of DrawingML-Specific Terms

**diagram** — A picture or graphical representation that is displayed using a related set of color, data, layout, and style parts. Examples of diagram types are cycle, organization chart, pyramid, target, and Venn.

## 14.2 Part Summary

The subclauses subordinate to this one describe in detail each of the part types specific to DrawingML.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Chart | WordprocessingML: Main  Document  SpreadsheetML: Drawings  PresentationML: Handout Master,  Notes Master, Notes Slide, Slide  Layout, Slide Master, Slide  All: Chart Drawing | chartSpace | §14.2.1 |
| Chart Drawing | All: Chart | userShapes | §14.2.2 |
| Diagram Colors | WordprocessingML: Main  Document  SpreadsheetML: Drawings  PresentationML: Handout Master,  Notes Master, Notes Slide, Slide  Layout, Slide Master, Slide | colorsDef | §14.2.3 |
| Diagram Data | WordprocessingML: Main  Document  SpreadsheetML: Drawings  PresentationML: Handout Master,  Notes Master, Notes Slide, Slide  Layout, Slide Master, Slide | dataModel | §14.2.4 |
| Diagram Layout Definition | WordprocessingML: Main  Document  SpreadsheetML: Drawings  PresentationML: Handout Master,  Notes Master, Notes Slide, Slide  Layout, Slide Master, Slide | layoutDef | §14.2.5 |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Diagram Style | WordprocessingML: Main  Document  SpreadsheetML: Drawings  PresentationML: Handout Master,  Notes Master, Notes Slide, Slide  Layout, Slide Master, Slide | styleDef | §14.2.6 |
| Theme | WordprocessingML: Main  Document  SpreadsheetML: Workbook  PresentationML: Handout Master,  Notes Master, Presentation, Slide Master | theme | §14.2.7 |
| Theme Override | PresentationML: Notes Slide, Slide, Slide Layout | themeOverride | §14.2.8 |
| Table Styles | PresentationML: Presentation | tblStyleLst | §14.2.9 |

*end note*]

#### 14.2.1 Chart Part

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

An instance of this part type describes a chart.

A package shall contain a Chart part for each chart in the document. In a WordprocessingML document, each such part shall be the target of an explicit relationship in a Main Document (§11.3.10) part. In a SpreadsheetML document, each such part shall be the target of an explicit relationship in a Drawings (§12.3.8) part. In a PresentationML document, each such part shall be the target of an explicit relationship in a Handout Master

(§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. This part is permitted to also be the target of an explicit relationship in a Chart Drawing (§14.2.2) part, if the chart that points at this Chart Drawing part is the target of a relationship from a Chartsheet part. In other words, the only time a chart can embed another chart is if the parent chart is part of a chartsheet.

[*Example*: The following Main Document part-relationship item contains relationships to two Chart parts, which are stored in the ZIP items ../charts/chart*N*.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/chart" Target="charts/chart1.xml"/>

<Relationship Id="rId5"

Type="http://…/chart" Target="charts/chart2.xml"/> </Relationships>

The following Drawings part-relationship item contains a relationship to a Chart part, which is stored in the ZIP item ../charts/chart1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/relationships/chart" Target="../charts/chart1.xml"/> </Relationships>

The following Slide part-relationship item contains relationships to two Chart parts, which are stored in the ZIP items ../charts/chart*N*.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/chart" Target="../charts/chart1.xml"/>

<Relationship Id="rId5"

Type="http://…/chart" Target="../charts/chart2.xml"/> </Relationships>

*end example*]

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

[*Example*: chart1.xml contains the following clustered bar chart:

<c:chartSpace …>

<c:chart>

<c:title>

…

</c:title>

<c:plotArea>

<c:layout>

…

</c:layout>

<c:barChart>

…

</c:barChart>

</c:plotArea>

<c:legend>

…

</c:legend>

</c:chart>

…

</c:chartSpace>

*end example*]

For WordprocessingML and PresentationML documents, the data for a chart is not stored in the Chart part directly. Instead, it shall be stored in an embedded SpreadsheetML package (§12.2) targeted by an Embedded Package (§15.2.11) part specified by that Chart part. For SpreadsheetML documents, the data for a chart is stored directly in the Drawing’s parent worksheet; no embedded SpreadsheetML package shall be used.

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

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

* Chart Drawing (§14.2.2)
* Embedded Package (§15.2.11)

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

#### 14.2.2 Chart Drawing Part

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

An instance of this part type contains all basic drawing elements (shapes) which are explicitly associated with this chart. These drawing elements are automatically moved with the chart when it is moved and resized when the chart is resized.

A package is permitted to contain one Chart Drawing part per chart part, and each such part shall be the target of an explicit relationship from a Chart (§14.2.1) part.

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

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/chartUserShapes" Target="../drawings/drawing1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<c:userShapes xmlns:cdr="…" xmlns:c="…">

<cdr:relSizeAnchor>

…

</cdr:relSizeAnchor> </c:userShapes>

*end example*]

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

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

 Chart (§14.2.1)

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

#### 14.2.3 Diagram Colors Part

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

An instance of this part type contains color information for a diagram.

A package shall contain exactly one Diagram Colors part per diagram. Each Diagram Colors part shall be the target of an explicit relationship in a WordprocessingML Main Document (§11.3.10), SpreadsheetML Drawings (§12.3.8), or PresentationML Slide (§13.3.8) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Colors parts, which are stored in the ZIP items ../graphics/colors*N*.xml.

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/diagramColors" Target="../graphics/colors1.xml"/>

<Relationship Id="rId8"

Type="http://…/diagramColors" Target="../graphics/colors2.xml"/> </Relationships>

*end example*]

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

[*Example*: colors1.xml contains the following:

<cs:colorsDef xmlns:cs="…" uniqueId="…" minVer="12.0">

<cs:title lang="" val="Primary Accent 2"/>

<cs:desc lang="" val="Primary Accent 2"/>

<cs:catLst>

<cs:cat type="accent1" pri="11200"/>

</cs:catLst>

<cs:styleLbl …>

…

</cs:styleLbl>

…

<cs:styleLbl …>

…

</cs:styleLbl> </cs:colorsDef>

*end example*]

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

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

#### 14.2.4 Diagram Data Part

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

An instance of this part type contains the semantic data for a diagram.

A package shall contain exactly one Diagram Data part per diagram. Each Diagram Data part shall be the target of an explicit relationship in a WordprocessingML Main Document (§11.3.10); a SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Data parts, which are stored in the ZIP items ../graphics/data*N*.xml.

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/diagramData" Target="../graphics/data1.xml"/> <Relationship Id="rId5"

Type="http://…/diagramData" Target="../graphics/data2.xml"/> </Relationships>

*end example*]

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

[*Example*: data1.xml contains the following:

<dm:dataModel xmlns:dm="…">

<dm:ptLst>

…

</dm:ptLst>

<dm:cxnLst>

…

</dm:cxnLst>

<dm:bg/>

<dm:whole/>

</dm:dataModel>

*end example*]

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

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

 Image (§15.2.14)

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

#### 14.2.5 Diagram Layout Definition Part

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

An instance of this part type is a template that describes how diagram-related data is mapped to a shape.

A package shall contain exactly one Diagram Layout Definition part per diagram. Each Layout Definition part shall be the target of an explicit relationship from a WordprocessingML Main Document (§11.3.10); a

SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. If a document contains multiple diagrams having the same graphic layout definition, each of those diagrams shall have its own copy of that Diagram Layout Definition part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Layout Definition parts, which are stored in the ZIP items ../graphics/layout*N*.xml.

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/diagramLayout" Target="../graphics/layout1.xml"/> <Relationship Id="rId6"

Type="http://…/diagramLayout" Target="../graphics/layout2.xml"/> </Relationships>

*end example*]

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

[*Example*: layout1.xml contains the following:

<lo:layoutDef xmlns:lo="…" uniqueId="…2" minVer="12.0" defStyle="">

<lo:title lang="" val="Hierarchy 2"/>

<lo:desc lang="" val=""/>

<lo:catLst>

<lo:cat type="hierarchy" pri="2000"/>

</lo:catLst>

<lo:sampData>

…

</lo:sampData>

<lo:styleData>

…

</lo:styleData>

<lo:clrData>

…

</lo:clrData>

<lo:layoutNode name="Name0" styleLbl="" moveWith="">

…

</lo:layoutNode> </lo:layoutDef>

*end example*]

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

A Diagram Layout Definition part is permitted to have explicit relationships to the following parts and items defined by ISO/IEC 29500:

 Image (§15.2.14)

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

#### 14.2.6 Diagram Style Part

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

An instance of this part type maps diagram semantic information to a document's theme.

A package shall contain exactly one Diagram Style part per diagram. Each Style part shall be the target of an explicit relationship from a WordprocessingML Main Document (§11.3.10); a SpreadsheetML Drawings part (§12.3.8); or a PresentationML Handout Master (§13.3.3), Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part.

[*Example*: The following SpreadsheetML Drawings part-relationship item contains a relationship to two Diagram Style parts, which are stored in the ZIP items ../graphics/quickStyle*N*.xml.

<Relationships xmlns="…">

<Relationship Id="rId3"

Type="http://…/diagramQuickStyle"

Target="../graphics/quickStyle1.xml"/>

<Relationship Id="rId7"

Type="http://…/diagramQuickStyle"

Target="../graphics/quickStyle2.xml"/>

</Relationships>

*end example*]

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

[*Example*: quickStyle1.xml contains the following:

<qs:styleDef xmlns:qs="…" uniqueId="…" minVer="12.0">

<qs:title lang="" val="Style 2"/>

<qs:desc lang="" val="Style 2"/>

<qs:catLst>

<qs:cat type="simple" pri="10200"/>

</qs:catLst>

<qs:scene3d>

…

</qs:scene3d>

<qs:style>

…

</qs:style>

<qs:styleLbl name="…">

…

</qs:styleLbl>

…

<qs:styleLbl name="…">

…

</qs:styleLbl> </qs:styleDef>

*end example*]

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

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

#### 14.2.7 Theme Part

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

An instance of this part type contains information about a document's *theme*, which is a combination of *color scheme*, *font scheme*, and *format scheme* (the latter also being referred to as *effects*). For a WordprocessingML document, the choice of theme affects the color and style of headings, among other things. For a SpreadsheetML document, the choice of theme affects the color and style of cell contents and charts, among other things. For a PresentationML document, the choice of theme affects the formatting of slides, handouts, and notes via the associated master, among other things.

A WordprocessingML or SpreadsheetML package shall contain zero or one Theme part, which shall be the target of an implicit relationship in a Main Document (§11.3.10) or Workbook (§12.3.23) part. A PresentationML package shall contain zero or one Theme part per Handout Master (§13.3.3), Notes Master (§13.3.4), Slide Master (§13.3.10) or Presentation (§13.3.6) part via an implicit relationship.

[*Example*: The following WordprocessingML Main Document part-relationship item contains a relationship to the Theme part, which is stored in the ZIP item theme/theme1.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/theme" Target="theme/theme1.xml"/> </Relationships>

*end example*]

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

[*Example*: theme1.xml contains the following, where the name attributes of the clrScheme, fontScheme, and fmtScheme elements correspond to the document's color scheme, font scheme, and format scheme, respectively:

<a:theme xmlns:a="…">

<a:themeElements>

<a:clrScheme name="…">

…

</a:clrScheme>

<a:fontScheme name="…">

…

</a:fontScheme>

<a:fmtScheme name="…">

…

</a:fmtScheme>

</a:themeElements>

<a:objectDefaults/> </a:theme>

*end example*]

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

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

 Image (§15.2.14)

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

#### 14.2.8 Theme Override Part

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

An instance of this part type contains information about an object’s *theme override*, which are overrides to the *color scheme*, *font scheme*, and *format scheme* (the latter also being referred to as *effects*) for a particular slide, notes slide, or handout.

A PresentationML package shall contain zero or one Theme Override part per Notes Slide (§13.3.5), Slide (§13.3.8), or Slide Layout (§13.3.9) part via an implicit relationship.

[*Example*: The following WordprocessingML Main Document part-relationship item contains a relationship to the Theme part, which is stored in the ZIP item theme/theme1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/themeOverride" Target="theme/themeoverride1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<a:ossOverride xmlns:a="…" >

…

</a:ossOverride>

*end example*]

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

A Theme Override part shall not contain implicit or explicit relationships to other parts defined by ISO/IEC 29500.

#### 14.2.9 Table Styles Part

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

An instance of this part type contains information about the table styles used by tables in this presentation. A table style defines characteristics such as row and column colors, heading row colors, and text.

A PresentationML package shall contain no more than one Table Styles part per Presentation (§13.3.6) part via an implicit relationship.

[*Example*: The following Presentation part-relationship item contains a relationship to a Table Styles part, which is stored in the ZIP item tableStyles.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/tableStyles" Target="tableStyles.xml"/> </Relationships>

*end example*]

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

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

<a:tblStyleLst xmlns:a="…">

<a:tblStyle>

…

</a:tblStyle>

</a:tblStyleLst>

*end example*]

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

A Table Styles part shall not contain implicit or explicit relationships to other parts defined by ISO/IEC 29500.

# 15. Shared

The relationship items and parts defined in this clause are used by one or more of WordprocessingML (§11), SpreadsheetML (§12), and PresentationML (§13) environments.

## 15.1 Glossary of Shared Terms

**array** — An array of mathematical elements (“e”) stacked vertically in a single math zone.

**build down** — The process of converting mathematical text from an implementation’s professional form to an implementation’s built-down form.

**build up** — The process of converting mathematical text from an implementation’s built-down form to an implementation’s professional form.

**built-down form** — An implementation-specific linear format that may or may not include additional rich formatting in addition to another plain-text linear format (such as TeX) or the linear format defined in Unicode Technical Note 28.

**built-up form** — See professional form. **control** — A region of active content within an Office Open XML document.

**display equation** — An equation that is in display mode, and thus is part of a display math zone. (Alternative names for display equation are: “display expression”, “display formula”, and “display math”.)

**display mode** — When mathematical text (i.e., text in one or more oMath blocks) is contained in a display math zone (i.e., an oMathPara block), the mathematical text represented in the oMath block(s) is in *display* *mode*. **equation array** — An array of equations. See array.

**inline equation** — An equation that is in an inline math zone. (Alternative names for inline equation are: “inline expression”, “inline formula”, and simply “inline math”.)

**instance of mathematical text** — A single continuous combination of mathematical text represented by a single oMath block and the OMML elements within that oMath block.

**linear format** — An implementation-specific plain-text 1-dimensional representation of mathematical text. **math accent** — A character that is specified as acceptable for use as an accent character by ISO/IEC 29500.

**math alphanumerics** — Characters with specific math styles, as defined in the Unicode Standard 5.0. **math paragraph** — One or more oMath elements (instances of mathematical text) that are in display mode. **math zone** — An isolated region of text within which mathematical text is used and outside of which mathematical text is not used. **mathematical text** — Any text meant to convey mathematical meaning through OMML.

**n-ary operator** — An operator that involves n terms when expanded. For instance, the following example uses the Unicode (ISO 10646) summation sign (U+2211) which has the official name “N-ARY SUMMATION”.

𝑛

∑ 𝑎𝑗 ≡ 𝑎1 + 𝑎2 + ⋯ + 𝑎𝑛

𝑗=1

**oMath container** — An oMath block that is part of a display math zone but is not itself a math zone.

**OMML** — Office Math Markup Language, a Shared ML of ISO/IEC 29500.

**professional form** — Implementation-specific 2-D representation of mathematical text. (Also referred to as "built-up form".)

## 15.2 Part Summary

The subclauses subordinate to this one describe in detail each of the shared part types.

[*Note*: For convenience, information from those subclauses is summarized in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Additional  Characteristics | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Characteristics | §15.2.1 |
| Audio | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Not applicable | §15.2.2 |
| Bibliography | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Sources | §15.2.3 |
| Custom XML Data Storage | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Not applicable | §15.2.4 |
| Custom XML Data Storage Properties | Custom XML Data Storage | datastoreItem | §15.2.6 |
| Digital Signature Origin | WordprocessingML,  SpreadsheetML, or  PresentationML package | Not applicable | §15.2.7 |
| Digital Signature XML Signature | Digital Signature Origin | Signature | §15.2.8 |
| Embedded Control | Numerous PresentationML, | Not applicable | §15.2.9 |
| **Part** | **Relationship Target of** | **Root Element** | **Ref.** |
| Persistence | SpreadsheetML, and  WordprocessingML parts |  |  |
| Embedded Object | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Not applicable | §15.2.10 |
| Embedded Package | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Not applicable | §15.2.11 |
| File Properties,  Extended | WordprocessingML,  SpreadsheetML, or  PresentationML package | Properties | § 15.2.12.3 |
| File Properties, Core | WordprocessingML,  SpreadsheetML, or  PresentationML package | coreProperties | §15.2.12.1 |
| File Properties,  Custom | WordprocessingML,  SpreadsheetML, or  PresentationML package | properties | §15.2.12.2 |
| Font | WordprocessingML Font  Table part, PresentationML  Presentation part | Not applicable | §15.2.13 |
| Image | Numerous PresentationML,  SpreadsheetML, and  WordprocessingML parts | Not applicable | §15.2.14 |
| Printer Settings | SpreadsheetML Chartsheet,  Dialogsheet, Worksheet parts,  WordprocessingML Main  Document or Glossary  Document parts | Not applicable | §15.2.15 |
| Thumbnail | WordprocessingML,  SpreadsheetML, or  PresentationML package | Not applicable | §15.2.16 |
| Video part | Numerous PresentationML and WordprocessingML parts | Not applicable | §15.2.17 |

*end note*]

#### 15.2.1 Additional Characteristics Part

|  |  |
| --- | --- |
| Content Type: | application/xml |
| Root  Namespace: | http://schemas.openxmlformats.org/officeDocument/2006/additionalCharacteristics |
| Source  Relationship: | http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml |

An instance of this part type contains information about additional characteristics of the producer that generated the document, when those characteristics cannot be specified using elements defined by ISO/IEC 29500. [*Note*: The contents of this part are purely informational, and do not place any requirements on subsequent consumption of the document. They are, however, intended to provide detailed information about the capabilities of the document’s producer, allowing those capabilities to be factored in during subsequent processing. For example, an application which supports 100,000 spreadsheet columns might choose to limit its output to 10,000 columns when presented with a document whose characteristics indicate that it was produced by an application with that limitation, in order prevent the introduction of content which is unsupported by the original producer (as that application might be used in the future to process this document). This markup is provided by ISO/IEC 29500 in order to provide an interoperable way of storing this information. *end note*]

A package is permitted to contain zero or one Additional Characteristics parts, and each such part shall be the target of an implicit relationship from a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Presentation (§13.3.6), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to an Additional Characteristics part, which is stored in the ZIP item ../customXML/item2.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/customXmlData" Target="../customXML/item2.xml"/> </Relationships>

*end example*]

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

An Additional Characteristics part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

 Custom XML Data Storage Properties (§15.2.6)

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

#### 15.2.2 Audio Part

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Content Type: | Any supported audio type.    [*Note*: Some example content types are: | | | |
|  | audio/aiff | [http://developer.apple.com/documentation/QuickTime/INMAC/SOUN](http://developer.apple.com/documentation/QuickTime/INMAC/SOUND/imsoundmgr.30.htm) |  |
|  |  |  | [D/imsoundmgr.30.htm](http://developer.apple.com/documentation/QuickTime/INMAC/SOUND/imsoundmgr.30.htm) |  |
| audio/midi | http://www.midi.org/about-midi/specinfo.shtml |
| audio/ogg | <http://xiph.org/vorbis/doc/Vorbis_I_spec.html> |
| audio/mpeg | ISO/IEC 11172-3 |
| *end note*] | |
| Root  Namespace: | not applicable | | | |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/audio | | | |

An instance of this part type contains an audio file.

A PresentationML package is permitted to contain zero or more Sound parts, each of which shall be the target of a relationship in a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part-relationship item. [*Example*: The following Slide partrelationship item contains a relationship to a Sound part, which is stored as the file E:\Beethoven's Symphony No. 9.wma:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/audio/x-ms-wma"

Target="file:///E:/Beethoven's%20Symphony%20No.%209.wma"

TargetMode="External"/> </Relationships>

*end example*]

An Audio part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

An Audio part is not stored as XML; instead, it involves a relationship target that is an audio clip.

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

A producer that wants interoperability should use the following standard format:

 audio/mpeg ISO/IEC 11172-3

#### 15.2.3 Bibliography Part

|  |  |
| --- | --- |
| Content Type: | application/xml |
| Root  Namespace: | http://purl.oclc.org/ooxml/officeDocument/bibliography |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/customXml |

An instance of this part type contains bibliographic data for the current package.

A package is permitted to contain zero or one Bibliography part, and each such part shall be the target of an implicit relationship in a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Bibliography part, which is stored in the ZIP item ../customXML/bib1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/customXml" Target="../customXML/bib1.xml"/> </Relationships>

*end example*]

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

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

 Custom XML Data Storage Properties (§15.2.6)

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

#### 15.2.4 Content Part

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Content Type: | Any supported XML content.    [*Note*: Some example content types are:   |  |  | | --- | --- | | image/svg+xml | <http://www.w3.org/TR/SVG11/> | | application/smil | <http://www.w3.org/TR/REC-smil/> | | text/xml | <http://www.w3.org/TR/MathML2/> |   *end note*]    If no explicit MIME type exists for a specific XML format, text/xml shall be used. Consumers who read a value of text/xml should determine the contents by the root namespace of the contents of the part. |
| Root  Namespace: | Various, as defined by the content type used. |
|  | [*Example*: MathML has a root namespace of [http://www.w3.org/1998/Math/MathML.](http://www.w3.org/1998/Math/MathML) *end example*] |
| Source  Relationship: | <http://purl.oclc.org/ooxml/officeDocument/relationships/customXml> |

An instance of this part type can contain XML markup of a format not defined by ISO/IEC 29500.

A package is permitted to contain zero or more Content parts, and each such part shall be the target of an explicit relationship from a Comments (§11.3.2), Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Glossary Document (§11.3.8), Header (§11.3.9), or Main Document (§11.3.10) part in a WordprocessingML package; a Drawings (§12.3.8) part in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or a Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Content part containing SVG markup, which is stored in the ZIP item svg1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/customXml" Target="../customXML/svg1.xml"/> </Relationships>

*end example*]

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

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

If a producer that wants interoperability supports equations, it should use one of the following standard formats:

* Office Open XML Math (§22.1
* W3C MathML 2.0

If a producer that wants interoperability supports ink annotations, it should use an ink annotation in this element in the following reference standard format:

* InkML http://www.w3.org/TR/inkregs

#### 15.2.5 Custom XML Data Storage Part

|  |  |
| --- | --- |
| Content Type: | application/xml |
| Root  Namespace: | any XML allowed |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/customXml |

An instance of this part type can contain arbitrary XML. As such, an instance of this part can be used to roundtrip arbitrary custom XML data with this package.

A package is permitted to contain one or more Custom XML Data Storage parts, and each such part shall be the target of an implicit relationship in a Main Document (§11.3.10) part in a WordprocessingML package; a Workbook (§12.3.23) part in a SpreadsheetML package; or a Handout Master (§13.3.3) , Notes Master (§13.3.4), Notes Slide (§13.3.5), Presentation (§13.3.6), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Main Document part-relationship item contains a relationship to a Custom XML Data Storage part, which is stored in the ZIP item ../customXML/item1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/customXmlData" Target="../customXML/item1.xml"/> </Relationships>

*end example*]

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

A Custom XML Data Storage part is permitted to have implicit relationships to the following parts defined by ISO/IEC 29500:

 Custom XML Data Storage Properties (§15.2.6)

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

#### 15.2.6 Custom XML Data Storage Properties Part

|  |  |
| --- | --- |
| Content Type: | application/vnd.openxmlformats-officedocument.customXmlProperties+xml |
| Root  Namespace: | http://purl.oclc.org/ooxml/officeDocument/customXmlDataProps |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/customXmlProps |

An instance of this part type contains the set of properties which are specified for this custom XML data. These properties consist of a unique ID for the storage, as well as information on the set of XML schemas used by this custom XML data storage.

A package is permitted to contain zero or more Custom XML Data Storage Properties parts, and each such part shall be the target of an implicit relationship from a Custom XML Data Storage (§15.2.4) part.

[*Example*: The following Custom XML Data Storage part-relationship item contains a relationship to a Custom XML Data Storage Properties part, which is stored in the ZIP item itemProps1.xml:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/customXmlProps" Target="itemProps1.xml"/> </Relationships>

*end example*]

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

[*Example*:

<ds:datastoreItem ds:itemID="{D85…53A}" xmlns:ds="…"/> \

*end example*]

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

A Custom XML Data Storage Properties part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

**15.2.7 Digital Signature Origin Part**

This part is defined in §13.2.1, “Digital Signature Origin Part”, of ISO/IEC 29500-2.

**15.2.8 Digital Signature XML Signature Part**

The part is defined in §13.2.2, “Digital Signature XML Signature Part”, of ISO/IEC 29500-2.

#### 15.2.9 Embedded Control Persistence Part

|  |  |
| --- | --- |
| Content Type: | Any supported control type.    [*Note*: There are a number of possible control types. One example of a potential control type would be an Active X control, which would use the following content type:  application/vnd.ms-office.activeX+xml. *end note*] |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/control |

An instance of this part contains information about an embedded control in the package. This information is provided by the specified control when asked to persist. [*Example*: An application might utilize the embedded object server technology KParts or Bonobo to store an embedded object using this part. *end example*]

A package is permitted to contain one or more Embedded Control Persistence parts, and each such part shall be the target of an explicit relationship in an Endnotes (§11.3.4), Footer (§11.3.6), Footnotes (§11.3.7), Header (§11.3.9), or Main Document (§11.3.10) part-relationship item in a WordprocessingML package; a Worksheet part (§12.3.24) in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part-relationship item in a PresentationML package.

The content type of this part shall determine the format and contents of the embedded control.

[*Example*: The following example shows the persistence that could be used for an embedded control which is a Java applet within a WordprocessingML document (the drawing object which provides a static image representation of the control, used when the Java applet itself is unavailable, has been omitted for brevity):

<w:p>

<w:r w:rsidR="005810E1">

<w:object w:dxaOrig="1440" w:dyaOrig="1440">

<w:drawing>

…

</w:drawing>

<w:control r:id="rId5" w:name="CommandButton1" w:shapeid="1027" />

</w:object>

</w:r>

</w:p>

The relationship type for rId5 is:

http://purl.oclc.org/ooxml/officeDocument/relationships/control

The XML content of the part referenced by rId5 could be:

<applet xlink:href="../../../../Program%20Files/Application" xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad" code="CalculateApplet.class" mayscript="false"/>

*end example*]

[*Example*: The following example shows the persistence that could be used for an embedded control which is an ActiveX control within a WordprocessingML document(the drawing object which provides a static image representation of the control, used when the ActiveX control itself is unavailable, has been omitted for brevity):

<w:p>

<w:r w:rsidR="005810E1">

<w:object w:dxaOrig="1440" w:dyaOrig="1440">

<w:drawing>

…

</w:drawing>

<w:control r:id="rId5" w:name="CommandButton1" w:shapeid="1027" />

</w:object>

</w:r>

</w:p>

The relationship type for rId5 is:

http://purl.oclc.org/ooxml/officeDocument/relationships/control

The content type of the part referenced by rId5 could be: application/vnd.ms-office.activeX+xml

The XML content of the part referenced by rId5 could be:

<ax:ocx ax:classid="{D7053240-CE69-11CD-A777-00DD01143C57}" ax:persistence="persistPropertyBag" xmlns:ax="http://schemas.microsoft.com/office/2006/activeX">

<ax:ocxPr ax:name="Caption" ax:value="CommandButton1" />

<ax:ocxPr ax:name="Size" ax:value="2540;847" />

<ax:ocxPr ax:name="FontName" ax:value="Calibri" />

<ax:ocxPr ax:name="FontHeight" ax:value="225" />

<ax:ocxPr ax:name="FontCharSet" ax:value="0" />

<ax:ocxPr ax:name="FontPitchAndFamily" ax:value="2" />

<ax:ocxPr ax:name="ParagraphAlign" ax:value="3" /> </ax:ocx>

*end example*]

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

An Embedded Control Persistence part shall not have any implicit or explicit relationships to other parts defined by ISO/IEC 29500.

#### 15.2.10 Embedded Object Part

|  |  |
| --- | --- |
| Content Type: | Any content type is allowed |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/oleObject |

An instance of this part type can contain an embedded object produced by any embedded object server.

A package is permitted to contain zero or more Embedded Object parts, and each such part shall be 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 in a WordprocessingML package; a Worksheet part (§12.3.24) in a SpreadsheetML package; or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part in a PresentationML package.

A WordprocessingML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Main Document part-relationship item. Each Embedded Object part shall have an associated image, which appears in the document as a placeholder for the corresponding embedded object.

[*Example*: Consider the case in which a WordprocessingML document has embedded in it one video object and one audio object. The following Main Document part-relationship item contains relationships to two Embedded parts (one each for the video and audio), which are stored in the ZIP items embeddings/embeddedObject*N*.bin:

<Relationships xmlns="…">

<Relationship Id="rId5"

Type="http://…/oleObject" Target="embeddings/embeddedObject1.bin"/> <Relationship Id="rId7"

Type="http://…/oleObject" Target="embeddings/embeddedObject2.bin"/>

<Relationship Id="rId4"

Type="http://…/image" Target="media/image1.png"/> <Relationship Id="rId6"

Type="http://…/image" Target="media/image2.png"/> </Relationships>

*example*]

A SpreadsheetML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Worksheet part-relationship item.

[*Example*: Consider the case in which a SpreadsheetML document has embedded in it one video object and one audio object on one worksheet, and another audio object embedded in another worksheet. The following Worksheet Document part-relationship item contains relationships to two Embedded Object parts (one each for the video and audio), which are stored in the ZIP items ../embeddings/embeddedObject*N*.bin:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/oleObject" Target="../embeddings/embeddedObject1.bin"/>

<Relationship Id="rId3"

Type="http://…/oleObject" Target="../embeddings/embeddedObject2.bin"/> </Relationships>

*end example*]

A PresentationML document package is permitted to contain zero or more Embedded Object parts, each of which shall be the target of a relationship in a Slide part-relationship item.

[*Example*: Consider the case in which a PresentationML document has embedded in it one video object and one audio object on one slide, and another audio object embedded on another slide. The following Slide partrelationship item contains relationships to two Embedded Object parts (one each for the video and audio), which are stored in the ZIP items ../embeddings/embeddedObject*N*.bin:

<Relationships xmlns="…">

<Relationship Id="rId6"

Type="http://…/oleObject"

Target="../embeddings/embeddedObject1.bin"/>

<Relationship Id="rId7"

Type="http://…/oleObject"

Target="../embeddings/embeddedObject2.bin"/> </Relationships>

*end example*]

An Embedded Object part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

An Embedded Object part is permitted to have an explicit relationship to the following parts defined by ISO/IEC 29500:

 Hyperlink (§15.3)

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

#### 15.2.11 Embedded Package Part

|  |  |
| --- | --- |
| Content Type: | Any content type is allowed |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/package |

An instance of this part type contains a complete package. For example, a WordprocessingML document might contain a SpreadsheetML or PresentationML document, in which case, the WordprocessingML document package would contain an embedded package part that defined that SpreadsheetML or PresentationML document.

A package is permitted to contain zero or more Embedded Package parts, and each such part shall be the target of an explicit relationship from a Chart (§14.2.1), 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 in a WordprocessingML package; a

Chart (§14.2.1), or Worksheet part (§12.3.24) in a SpreadsheetML package; or a Chart (§14.2.1), Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), Slide Master (§13.3.10) part in a PresentationML package.

[*Example*: The following Presentation part-relationship item contains relationships to two Embedded Package parts: one is a SpreadsheetML package, which is stored in the ZIP item embeddings/Worksheet1.xlsx, the other is a PresentationML package, which is stored in the ZIP item embeddings/Presentation2.pptx. The image files are used as document display placeholders if the consumer cannot handle the embedded package type:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/image" Target="media/image1.emf"/>

<Relationship Id="rId5"

Type="http:package" Target="embeddings/Worksheet1.xlsx"/>

<Relationship Id="rId6"

Type="http://…/image" Target="media/image2.emf"/>

<Relationship Id="rId7"

Type="http://…/package" Target="embeddings/Presentation2.pptx"/> </Relationships>

*end example*]

An Embedded Package part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

An Embedded Package part is permitted to have an explicit relationship to the following parts defined by ISO/IEC 29500:

 Hyperlink (§15.3)

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

#### 15.2.12 File Properties

There are three kinds of file properties: *core*, *custom*, and *extended*. 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. *Extended file properties* are specific to Office Open XML packages, while *custom file properties* are defined by the user, with each custom file property having a name, a value, and a type.

##### 15.2.12.1 Core File Properties Part

This part and the related OPC part is defined in §11, “Core Properties”, of ISO/IEC 29500-2. 15.2.12.2 Custom File Properties Part

|  |  |
| --- | --- |
| Content Type: | application/vnd.openxmlformats-officedocument.custom-properties+xml |
| Root  Namespace: | http://purl.oclc.org/ooxml/officeDocument/customProperties |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/customProperties |

An instance of this part contains the names of custom file properties that apply to the package, their values, and the types of those values. A custom file property might be 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.

A package shall contain at most one Custom File Properties part, and that part shall be the target of a relationship in the package-relationship item for the document.

[*Example*: The following PresentationML's package-relationship item contains a relationship to a Custom File Properties part, stored in the ZIP item docProps/custom.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/custom-properties" Target="docProps/custom.xml"/> </Relationships>

*end example*]

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

[*Example*: Here's some content markup from a WordprocessingML document, which contains four custom properties: Client, having a text value of "ACME Corp."; Document number, having a numeric value of 1543; Recorded date, having a date/time value of 2005-12-01; and Special processing needed, having a Boolean value of false:

<Properties … xmlns:vt="…">

<property fmtid="{D5C…9AE}" pid="2" name="Client">

<vt:lpwstr>ACME Corp.</vt:lpwstr>

</property>

<property fmtid="{D5C…9AE}" pid="3" name="Document number">

<vt:i4>1543</vt:i4>

</property>

<property fmtid="{D5C…9AE}" pid="4" name="Recorded date">

<vt:filetime>2005-12-01T05:00:00Z</vt:filetime>

</property>

<property fmtid="{D5C…9AE}" pid="5" name="Special processing needed">

<vt:bool>false</vt:bool>

</property>

</Properties>

*end example*]

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

A Custom File Properties part shall not have implicit or explicit relationships to other parts defined by ISO/IEC 29500.

##### 15.2.12.3 Extended File Properties Part

|  |  |
| --- | --- |
| Content Type: | application/vnd.openxmlformats-officedocument.extended-properties+xml |
| Root  Namespace: | http://purl.oclc.org/ooxml/officeDocument/extendedProperties |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/extendedProperties |

An instance of this part contains properties specific to an Office Open XML document. [*Example*: A

PresentationML document specifies the number of slides in this presentation when last saved by a producer. *end example*]

A package shall contain at most one Extended File Properties part, and that part shall be the target of a relationship in the package-relationship item for the document.

[*Example*:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/extended-properties" Target="docProps/app.xml"/> </Relationships>

*end example*]

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

[*Example*: Here's some content markup from a WordprocessingML document:

<Properties …>

<Template>Normal.dotm</Template>

<TotalTime>0</TotalTime>

<Pages>1</Pages>

<Words>3</Words>

<Characters>22</Characters>

<Application>Sample Producer</Application>

<DocSecurity>0</DocSecurity>

<Lines>1</Lines>

<Paragraphs>1</Paragraphs>

…

<AppVersion>12.0000</AppVersion> </Properties>

here's some content markup from a SpreadsheetML document:

<Properties …>

<Application>Sample Producer</Application>

<HeadingPairs>

…

</HeadingPairs>

<TitlesOfParts>

…

</TitlesOfParts>

<Company>Consultant</Company>

…

</Properties>

and here's some content markup from a PresentationML document:

<Properties …>

<Template>ppt\_template\_sdwest05</Template>

<TotalTime>3166</TotalTime>

<Words>37</Words>

<Application>Sample Producer</Application>

<PresentationFormat>On-screen Show</PresentationFormat>

<Paragraphs>15</Paragraphs>

<Slides>2</Slides>

<Notes>2</Notes>

…

<HeadingPairs>

…

</HeadingPairs>

<TitlesOfParts>

…

</TitlesOfParts>

…

</Properties>

*end example*]

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

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

#### 15.2.13 Font Part

|  |  |
| --- | --- |
| Content Type: | application/x-fontdata application/x-font-ttf  application/vnd.openxmlformats-officedocument.obfuscatedFont |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/font |

An instance of this part type contains a given font embedded directly into the document. (This is useful when using custom fonts or fonts that are not widely distributed.)

Fonts stored in a Font part can be stored in one of the following formats, identified by the associated content type:

* application/x-fontdata specifies that the font shall be stored in the Embedded OpenType Format of http://www.w3.org/Submission/2008/SUBM-EOT-20080305
* application/x-font-ttf specifies that the font shall be stored in a format conforming to Open Font Structure defined in ISO/IEC 14496-22:2008 §3.5. [*Note*: The TrueType Collection format defined in ISO/IEC 14496-22:2008 §3.6 cannot be used. *end note*]
* application/vnd.openxmlformats officedocument.obfuscatedFont specifies that the font is obfuscated using the algorithm specified by Font Embedding (§17.8.1). The source font shall be stored in a format conforming to Open Font Structure defined in ISO/IEC 14496-22:2008 §3.5. [*Note*: The TrueType Collection format defined in ISO/IEC 14496-22:2008 §3.6 cannot be used. *end note*] Only packages of type WordprocessingML are permitted to reference this content type.

If a font is stored in the ISO/IEC 14496-22:2007 format, it shall only be used when stored as an individual font. [*Note:* Font collections should be converted into individual fonts before they are embedded using this part. *end note*]

A package shall contain zero or more Font parts, and for each that exists, that part shall be the target of an explicit relationship in the Font Table (§11.3.5), or Presentation (§13.3.6) part.

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

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

#### 15.2.14 Image Part

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Content Type: | Any supported image type.    [*Note*: Some example content types are:   |  |  | | --- | --- | | image/gif | http://www.w3.org/Graphics/GIF/spec-gif89a.txt | | image/png | ISO/IEC 15948:2003 http://www.libpng.org/pub/png/spec/ | | image/tiff | http://partners.adobe.com/public/developer/tiff/index.html#spec | | image/pict | [http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-](http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html)  [2.html](http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html) | | image/jpeg | http://www.w3.org/Graphics/JPEG/ |   e*nd note*] |
| Root  Namespace: | Not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/image |

An image can be stored in a package as a ZIP item. Image ZIP items shall be identified by an image part relationship and the appropriate content type.

A package is permitted to contain zero or more Image parts, and each such part shall be 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), Drawing (§12.3.8), or Main Document (§11.3.10) part in a WordprocessingML package or a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10 part in a PresentationML package.

[*Example*: The following PresentationML's package-relationship item contains one relationship, for the slide template jpeg image stored in the ZIP item ../media/image1.jpeg:

<Relationships xmlns="…">

<Relationship Id="rId8"

Type="http://…/image" Target="../media/image1.jpeg"/> </Relationships>

*end example*]

An Image part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

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

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

* image/png ISO/IEC 15948:2003, http://www.libpng.org/pub/png/spec/
* image/jpeg, http://www.w3.org/Graphics/JPEG

#### 15.2.15 Printer Settings Part

|  |  |
| --- | --- |
| Content Type: | application/vnd.openxmlformats-officedocument.spreadsheetml.printerSettings (in SpreadsheetML documents)    application/vnd.openxmlformats-officedocument.wordprocessingml.printerSettings (in WordprocessingML documents)    application/vnd.openxmlformats-officedocument.presentationml.printerSettings (in PresentationML documents) |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/printerSettings |

An instance of this part type contains information about the initialization and environment of a printer or a display device. The layout of this information is application-defined.

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

[*Example*: An Office Open XML producer on Windows might store the DEVMODE structure defined here:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/prntspol\_8nle.asp,](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/prntspol_8nle.asp) while an application on the Mac OS might choose to store the print record defined here:

[http://developer.apple.com/documentation/Printing/index.html.](http://developer.apple.com/documentation/Printing/index.html) *end example*]

A SpreadsheetML package is permitted to contain at most one Printer Settings part per Chartsheet, Dialogsheet, or Worksheet part, and that part shall be the target of an implicit relationship from a Chartsheet (§12.3.2), Dialogsheet (§12.3.7), or Worksheet (§12.3.24) part. A WordprocessingML package is permitted to contain zero or more Printer Settings parts, one per sectPr element, each a target of an explicit relationship from a Main Document (§11.3.10) or Glossary Document (§11.3.8) part. A PresentationML package is permitted to contain at most one Printer Settings part, and that part shall be the target of an implicit relationship from a Presentation (§13.3.6) part.

[*Example*: The following SpreadsheetML Worksheet part-relationship item contains a relationship to a Printer Settings part, which is stored in the ZIP item ../printerSettings/printerSettings1.xml:

<Relationships xmlns="…">

<Relationship Id="rId4"

Type="http://…/printerSettings"

Target="../printerSettings/printerSettings1.xml"/> </Relationships>

where the contents of PrinterSettings1.xml contains the following XML:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<PrinterSettings xmlns="…">

<PrinterSetting name="PropertyName" value="PropertyValue" /> </PrinterSettings>

*end example*]

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

#### 15.2.16 Thumbnail Part

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Content Type: | Any supported image type.    [*Note*: Some example content types are:   |  |  | | --- | --- | | image/gif | http://www.w3.org/Graphics/GIF/spec-gif89a.txt | | image/png | ISO/IEC 15948:2003 http://www.libpng.org/pub/png/spec/ | | image/tiff | http://partners.adobe.com/public/developer/tiff/index.html#spec | | image/pict | [http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-](http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html)  [2.html](http://developer.apple.com/documentation/mac/QuickDraw/QuickDraw-2.html) | | image/jpeg | http://www.w3.org/Graphics/JPEG/ |   e*nd note*] |
| Root  Namespace: | Not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/metadata/thumbnail |

To help end-users identify parts of a package or the package as a whole, images, called *thumbnails*, can be stored in that package. Each thumbnail image is generated by the package producer and is stored in the package as a ZIP item. There are no limitations on the size or dimensions of the thumbnail produced, and applications are free to scale the images as desired.

Thumbnail ZIP items shall be identified by either a package-relationship item or a part-relationship item. Packages shall not contain more than one thumbnail relationship associated with the package as a whole, or more than one thumbnail relationship per package part.

[*Example*: The following PresentationML's package-relationship item contains one relationship, for the metafile image stored in the ZIP item thumbnail.wmf:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/thumbnail" Target="docProps/thumbnail.wmf"/> </Relationships>

*end example*]

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

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

#### 15.2.17 Video Part

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Content Type: | Any supported video type.    [*Note*: Some example content types are:   |  |  | | --- | --- | | video/avi | http://www.the-labs.com/Video/odmlff2-avidef.pdf | | video/mpg | ISO/IEC 13818 | | video/mpeg | ISO/IEC 13818 | | video/ogg | <http://www.theora.org/doc/Theora.pdf> | | video/quicktime | <http://developer.apple.com/documentation/QuickTime/> | | video/vc1 | <http://tools.ietf.org/html/rfc4425> |   *end note*] |
| Root  Namespace: | not applicable |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/video |

An instance of this part type contains a video file.

A PresentationML package is permitted to contain zero or more Video parts, each of which shall be the target of an explicit relationship in a Handout Master (§13.3.3), Notes Slide (§13.3.5), Notes Master (§13.3.4), Slide (§13.3.8), Slide Layout (§13.3.9), or Slide Master (§13.3.10) part. A WordprocessingML package is permitted to contain zero or more Video parts, each of which shall be 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.

[*Example*: The following Slide part-relationship item contains a relationship to a Video part, which is stored as the file E:\Video demo.avi:

<Relationships xmlns="…">

<Relationship Id="rId2"

Type="http://…/video"

Target="file:///E:\Video%20demo.avi" TargetMode="External"/> </Relationships>

*end example*]

A Video part is not stored as XML; instead, it involves a relationship target that is a video clip.

A Video part can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

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

A producer that wants interoperability should use the following standard format:

 video/mpeg ISO/IEC 13818

## 15.3 Hyperlinks

|  |  |
| --- | --- |
| Source  Relationship: | http://purl.oclc.org/ooxml/officeDocument/relationships/hyperlink |

A hyperlink can be stored in a package as a relationship. Hyperlinks shall be identified by containing a target which specifies the destination of the given hyperlink.

[*Example*: The following WordprocessingML Footnote part's relationship part contains one relationship, for the hyperlink http://schemas.openxmlformats.org/wordprocessingml/:

<Relationships xmlns="…">

<Relationship Id="rId1"

Type="http://…/hyperlink"

Target="http://schemas.openxmlformats.org/wordprocessingml/"

TargetMode="External"/> </Relationships>

*end* example]

A hyperlink target can be located within or external to the package containing the relationships part (expressed syntactically, the TargetMode attribute of the Relationship element can be Internal or External).

# 16. Part Overview

**This clause is informative.**

For convenience, the following subclauses specify the root elements for each part and, when applicable, link to the appropriate subclause that defines the root element for that part within the package. [*Note*: The terms part and package, as used here, are defined in ISO/IEC 29500-2. *end note*] Note that the contents of some of these parts are not defined in this Part of ISO/IEC 29500, in which case, the root element is listed as "Not applicable" and the Reference is "n/a". For those parts where the Root Element is listed as "Not Applicable", a definition of the part’s use and how its content shall be structured is defined in this Part 1 of ISO/IEC 29500.

## 16.1 WordprocessingML Summary

|  |  |  |
| --- | --- | --- |
| **Part** | **Root Element** | **Ref.** |
| Alternative Format Import | Not applicable | n/a |
| Comments | comments | §17.13.4.6 |
| Document Settings | settings | §17.15.1.78 |
| Endnotes | endnotes | §17.11.8 |
| Font Table | fonts | §17.8.3.11 |
| Footer | ftr | §17.10.3 |
| Footnotes | footnotes | §17.11.15 |
| Glossary Document | glossaryDocument | §17.12.10 |
| Header | hdr | §17.10.4 |
| Mail Merge Recipient Data | recipients | §17.14.29 |
| Main Document | document | §17.2.3 |
| Numbering Definitions | numbering | §17.9.16 |
| Style Definitions | styles | §17.7.4.18 |
| Web Settings | webSettings | §17.15.2.45 |

## 16.2 SpreadsheetML Summary

|  |  |  |
| --- | --- | --- |
| **Part** | **Root Element** | **Ref.** |
| Calculation Chain | calcChain | §18.6.2 |
| Chartsheet | chartsheet | §18.3.1.12 |
| Comments | comments | §18.7.6 |
| Connections | connections | §18.13.2 |
| Custom Property | Not applicable | n/a |
| **Part** | **Root Element** | **Ref.** |
| Custom XML Mappings | MapInfo | §18.16.3 |
| Dialogsheet | dialogsheet | §18.3.1.34 |
| Drawing | wsDr | §20.5.2.35 |
| External Workbook References | externalLink | §18.14.8 |
| Metadata | metadata | §18.9.8 |
| Pivot Table | pivotTableDefinition | §18.10.1.73 |
| Pivot Table Cache Definition | pivotCacheDefinition | §18.10.1.67 |
| Pivot Table Cache Records | pivotCacheRecords | §18.10.1.68 |
| Query Table | queryTable | §18.12.2 |
| Shared String Table | sst | §18.4.9 |
| Shared Workbook Revision Headers | headers | §18.11.1.1 |
| Shared Workbook Revision Log | revisions | §18.11.1.16 |
| Shared Workbook User Data | users | §18.11.2.2 |
| Single Cell Table Definitions | singleXmlCells | §18.5.2.2 |
| Styles | styleSheet | §18.8.39 |
| Table Definition | table | §18.5.1.2 |
| Volatile Dependencies | volTypes | §18.15.6 |
| Workbook | workbook | §18.2.27 |
| Worksheet | worksheet | §18.3.1.99 |

## 16.3 PresentationML Summary

|  |  |  |
| --- | --- | --- |
| **Part** | **Root Element** | **Ref.** |
| Comment Authors | cmAuthorLst | §19.4.3 |
| Comments | cmLst | §19.4.4 |
| Handout Master | handoutMaster | §19.3.1.24 |
| Notes Master | notesMaster | §19.3.1.27 |
| Notes Slide | notes | §19.3.1.26 |
| Presentation | presentation | §19.2.1.26 |
| Presentation Properties | presentationPr | §19.2.1.27 |
| Slide | sld | §19.3.1.38 |
| Slide Layout | sldLayout | §19.3.1.39 |
| Slide Master | sldMaster | §19.3.1.42 |
| Slide Synchronization Data | sldSyncPr | §19.6.1 |
| User-Defined Tags | tagLst | §19.3.3.2 |
| **Part** | **Root Element** | **Ref.** |
| View Properties | viewPr | §19.2.2.18 |

## 16.4 DrawingML Summary

|  |  |  |
| --- | --- | --- |
| **Part** | **Root Element** | **Ref.** |
| Chart | chartSpace | §21.2.2.29 |
| Chart Drawing | userShapes | §21.2.2.220 |
| Diagram Colors | colorsDef | §21.4.4.3 |
| Diagram Data | dataModel | §21.4.2.10 |
| Diagram Layout Definition | layoutDef | §21.4.2.16 |
| Diagram Style | styleDef | §21.4.5.7 |
| Theme | theme | §20.1.6.9 |
| Theme Override | themeOverride | §20.1.6.12 |
| Table Styles | tblStyleLst | §20.1.4.2.27 |

## 16.5 Shared Summary

|  |  |  |
| --- | --- | --- |
| **Part** | **Root Element** | **Ref.** |
| Additional Characteristics | additionalCharacteri stics | §22.7.2.1 |
| Audio | Not applicable | n/a |
| Bibliography | Sources | §22.6.2.60 |
| Custom XML Data Storage | Not applicable | n/a |
| Custom XML Data Storage Properties | datastoreItem | §22.5.2.1 |
| Digital Signature Origin | Not applicable | n/a |
| Digital Signature XML Signature | Signature | Defined in  ISO/IEC  29500-2 |
| Embedded Control Persistence | Not applicable | n/a |
| Embedded Object | Not applicable | n/a |
| Embedded Package | Not applicable | n/a |
| File Properties, Core | coreProperties | Defined in  ISO/IEC  29500-2 |
| File Properties, Custom | Properties | §22.3.2.1 |
| File Properties, Extended | Properties | §22.2.2.21 |
| Font | Not applicable | n/a |
| **Part** | **Root Element** | **Ref.** |
| Image | Not applicable | n/a |
| Printer Settings | Not applicable | n/a |
| Thumbnail | Not applicable | n/a |
| Video | Not applicable | n/a |

**End of informative text.**