Form 36

DPC: 25/30464096 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000
Fax: +44 (0)20 8996 7400
www.bsigroup.com

Date: 21 August 2025
Origin: International

Latest date for receipt of comments: 14 October 2025

Project No. 2022/02363

Responsible committee: IST/41 Document description and processing language

Interested committees:

Title: Draft BS ISO/IEC 29500-1 Document description and processing languages - Office Open XML File Formats. Part 1: Fundamentals and Markup Language Reference.

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of the content of this standard:

i) has any issues related to 3rd party IPR, patent or copyright
ii) affects other national standard(s)
iii) requires additional national guidance or information

WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD. THIS DRAFT IS NOT CURRENT BEYOND 14 October 2025

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop http://shop.bsigroup.com or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from https://standardsdevelopment.bsigroup.com/.

Documents
Document Sharing Hub

Full document details: 25/30464096 DC

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at **https://standardsdevelopment.bsigroup.com/**. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex **(e.g. 3.1)**	Paragraph/ Figure/Table/Note	**Type of comment**	**Commend (justification for change) by the MB**	**Proposed change by the MB**	**Secretariat observations on each comment submitted**
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	

AVA Documents
Document Sharing Hub

Full document details: 25/30464096 DC

DRAFT International Standard

ISO/IEC DIS 29500-1

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

Part 1: Fundamentals and Markup Language Reference

ICS: 35.240.30; 35.060

ISO/IEC JTC **1**/SC **34**

Secretariat: **JISC**

Voting begins on:
2025-08-21

Voting terminates on:
2025-11-13

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 29500-1:2025(en)

© ISO/IEC 2025

Documents
Document Sharing Hub

Full document details: 25/30464096 DC



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, 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
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Documents
Document Sharing Hub

Full document details: 25/30464096 DC

Contents

Foreword ..... **vii**

Introduction ..... **ix**

1. Scope **1**

2. Normative references **1**

3. Terms and definitions **4**

3.1 General 4
3.2 WordprocessingML-specific 7
3.3 SpreadsheetML-specific 8
3.4 PresentationML-specific 9
3.5 DrawingML-specific 10
3.6 Shared 10

4. Notational conventions **12**

5. Organizational conventions **12**

6. External data files **12**

7. Conformance **13**

7.1 Document conformance 13
7.2 Application conformance 13
7.3 Application descriptions 14
7.4 Interoperability guidelines 15

8. Overview **16**

8.1 Content overview 16
8.2 Packages and parts 16
8.3 Consumers and producers 16
8.4 WordprocessingML 16
8.5 SpreadsheetML 17
8.6 PresentationML 18
8.7 Supporting MLs 18

9. Packages **20**

9.1 General 20
9.2 Office Open XML's use of OPC 20
9.3 Relationships in Office Open XML 21

10. Markup compatibility and extensibility **25**

10.1 General 25
10.2 Validity and MCE 25
10.3 Applications and MCE 27

11. WordprocessingML **29**

11.1 General 29
11.2 Package structure 29
11.3 Parts 31
11.4 Document template 53
11.5 Framesets 54
11.6 Master documents and subdocuments 54

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

11.7 Mail merge data source 55
11.8 Mail merge header data source 56
11.9 XSL transformation 57

12. SpreadsheetML 58
12.1 General 58
12.2 Package structure 58
12.3 Parts 60
12.4 External workbooks 89

13. PresentationML 90
13.1 General 90
13.2 Package structure 90
13.3 Parts 92
13.4 HTML publish location 107
13.5 Slide synchronization server location 108

14. DrawingML 108
14.1 General 108
14.2 Parts 108

15. Shared 119
15.1 General 119
15.2 Parts 119
15.3 Hyperlinks 135

16. Part overview 136
16.1 General 136
16.2 WordprocessingML 136
16.3 SpreadsheetML 137
16.4 PresentationML 138
16.5 DrawingML 139
16.6 Shared 140

17. WordprocessingML reference material 141
17.1 Table of contents and introduction 141
17.2 Main document story 198
17.3 Paragraphs and rich formatting 204
17.4 Tables 370
17.5 Custom markup 471
17.6 Sections 520
17.7 Styles 588
17.8 Fonts 631
17.9 Numbering 650
17.10 Headers and footers 686
17.11 Footnotes and endnotes 697
17.12 Glossary document 727
17.13 Annotations 740
17.14 Mail merge 863
17.15 Settings 895
17.16 Fields and hyperlinks 1066
17.17 Miscellaneous topics 1185
17.18 Simple types 1191

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

18. SpreadsheetML reference material 1402
18.1 Table of contents and introduction 1402
18.2 Workbook 1442
18.3 Worksheets 1485
18.4 Shared string table 1598
18.5 Tables 1604
18.6 Calculation chain 1619
18.7 Comments 1620
18.8 Styles 1625
18.9 Metadata 1671
18.10 Pivot tables 1684
18.11 Shared workbook data 1817
18.12 QueryTable data 1842
18.13 External data connections 1848
18.14 Supplementary workbook data 1867
18.15 Volatile dependencies 1874
18.16 Custom XML mappings 1877
18.17 Formulas 1883
18.18 Simple types 2183

19. PresentationML reference material 2259
19.1 Table of contents and introduction 2259
19.2 Presentation 2268
19.3 Slides 2299
19.4 Comments 2333
19.5 Animation 2337
19.6 Slide synchronization data 2418
19.7 Simple types 2418

20. DrawingML - Framework reference material 2448
20.1 DrawingML - Main 2448
20.2 DrawingML - Picture 2761
20.3 DrawingML - Locked canvas 2768
20.4 DrawingML - WordprocessingML Drawing 2768
20.5 DrawingML - SpreadsheetML Drawing 2820

21. DrawingML - Components reference material 2845
21.1 DrawingML - Main 2845
21.2 DrawingML - Charts 3014
21.3 DrawingML - Chart Drawings 3119
21.4 DrawingML - Diagrams 3137

22. Shared MLs reference material 3237
22.1 Math 3237
22.2 Extended properties 3342
22.3 Custom properties 3347
22.4 Variant types 3349
22.5 Custom XML data properties 3359
22.6 Bibliography 3362
22.7 Additional characteristics 3390
22.8 Office document relationships 3394
22.9 Shared simple types 3394

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

23. Custom XML schema references ... 3411
23.1 Table of contents and introduction ... 3411
23.2 Elements ... 3412

Annex A. (normative) Schemas – W3C XML Schema ... 3415

Annex B. (informative) Schemas – RELAX NG ... 3416

Annex C. (normative) WordprocessingML page borders ... 3418

Annex D. (normative) Predefined SpreadsheetML style definitions ... 3419
D.1 General ... 3419
D.2 Built-in table styles ... 3419
D.3 Built-in cell styles ... 3475
D.4 Built-in PivotTable AutoFormats ... 3479

Annex E. (informative) Example predefined DrawingML shape and text geometries ... 3495

Annex F. (informative) Bidirectional support ... 3496
F.1 Introduction ... 3496
F.2 Shared (WordprocessingML and DrawingML) ... 3496
F.3 WordprocessingML ... 3498
F.4 SpreadsheetML ... 3501
F.5 PresentationML ... 3502
F.6 DrawingML ... 3502
F.7 The Unicode bidirectional algorithm and Office Open XML ... 3502

Annex G. (informative) Accessibility best practices ... 3505
G.1 General ... 3505
G.2 The value of creating an accessible Office Open XML implementation ... 3505
G.3 Needs by type of disability ... 3505
G.4 Best practices for developers ... 3508
G.5 Best practices for document and template authors ... 3511
G.6 Best practices for customers of Office Open XML implementations ... 3521

Annex H. (informative) WordprocessingML page layout concepts ... 3532
H.1 Physical settings and logical settings ... 3532
H.2 Margin, paper size, column, and gap ... 3532
H.3 Document grid for East Asian languages ... 3537
H.4 Using the document grid for character and line positioning for East Asian languages ... 3540

Annex I. (informative) DrawingML coordinate systems and transformations ... 3541
I.1 Introduction ... 3541
I.2 Coordinate System ... 3541
I.3 Shape Transformations ... 3541
I.4 Group Transformations ... 3544
I.5 Nesting Transformations ... 3548
I.6 Transformation Matrices ... 3548

Bibliography ... 3550

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, Document description and processing languages.

This fifth edition cancels and replaces the fourth edition (ISO/IEC 29500-1:2016), which has been technically revised.

The main changes are as follows:

- Corrections and clarifications have been made throughout the text, but no new functionality has been added.
- Editorial changes have been made throughout the text to conform to ISO stylistic requirements:
  - Headings use sentence case, not title case.
  - Nested tables have been extracted.
  - Previously used Example, Note, Rationale, and Guidance tags have been condensed to Examples and Notes only and reformatted.
  - Figure and table names and numbers have been added.
  - Hanging paragraphs have been moved to new introductory subclauses at the top of the major collecting clause or subclause, which caused renumbering of some subclauses.
  - The introductory clauses have been reordered.
- Where appropriate, normative and bibliographical references have been added, removed, or updated to use undated or more recent versions.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

- Uses of BNF or EBNF have been changed to ABNF.
- Clarifications and corrections have been made to separations between Strict and Transitional and some content has been moved from this document to ISO/IEC 29500-4.
- Clause 3 (Terms and definitions) has been revised by reorganizing terms into groups.
- Subclauses containing ML-specific glossaries of terms have been integrated into Clause 3.
- Clause 10 (Markup Compatibility and Extensibility) has been expanded and revised in its entirety.
- The printed schemas in Annexes A & B have been removed; schemas exist in digital form only.
- The following annexes have been removed:
    - Additional Syntax Constraints
    - Namespace Prefix Mapping in Examples
    - WordprocessingML Custom XML Data Extraction
    - Root Element Locations (This information has been merged into clause 16.)
    - Primer (Some information has been integrated into the main text or annexes.)
    - Differences Between ISO/IEC 29500 and ECMA-376:2006
- The following annexes have been added:
    - WordprocessingML page layout
    - DrawingML coordinate systems and transformations

A list of all parts in the ISO/IEC 29500 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

Introduction

The ISO/IEC 29500 series 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[1] documents.

[1] This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

Documents
Document Sharing Hub

Document description and processing languages — Office Open XML File Formats

Part 1: Fundamentals and Markup Language Reference

1. Scope

Together, this document and ISO/IEC 29500-4 define a set of XML vocabularies for representing word-processing documents, spreadsheets and presentations. The goals of ISO/IEC 29500 are to be capable of faithfully representing the pre-existing corpus of word-processing documents, spreadsheets and presentations that had been produced by the Microsoft Office applications, from Microsoft Office 97 to Microsoft Office 2008, inclusive, at the date of creation of ISO/IEC 29500 and to facilitate extensibility and interoperability by enabling implementations by multiple vendors and on multiple platforms.

This document specifies the features of Office Open XML documents that, in their semantics, syntax, and packaging, satisfy the requirements of conformance class Office Open XML Strict, as specified in Clause 7. This document also specifies requirements for Office Open XML consumers and producers.

2. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

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

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

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

ISO/IEC 29500-2:2021, *Document description and processing languages — Office Open XML File Formats – Part 2: Open Packaging Conventions*

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

IEC 60559:2020, *Information technology — Microprocessor Systems — Floating-Point arithmetic*

IEC 61966-2-1, *Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB*

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

IANA (INTERNET ASSIGNED NAMES AUTHORITY). *Media Types*. Online. Edited by A. Melnikov, D. Miller, M. Kucherawy. Available from: https://www.iana.org/assignments/media-types/media-types.xhtml.

IANA. *Character Set Registrations*. Online. Edited by M. Dürst. Available from: https://www.iana.org/assignments/charset-reg/charset-reg.xhtml.

IANA. *Character Sets*. Online. Edited by M. Dürst. Available from: https://www.iana.org/assignments/character-sets/character-sets.xhtml.

IETF. BCP 47. Online. Available from: https://datatracker.ietf.org/doc/bcp47/.

IETF. RFC 1319: *The MD2 Message-Digest Algorithm*. Online. Edited by B. Kaliski. April 1992. Available from: https://datatracker.ietf.org/doc/rfc1319/.

IETF. RFC 1320: *The MD4 Message-Digest Algorithm*. Online. Edited by R. Rivest. April 1992. Available from: https://datatracker.ietf.org/doc/rfc1320/https://www.rfc-editor.org/rfc/rfc1320.

IETF. RFC 1321: *The MD5 Message-Digest Algorithm*. Online. Edited by R. Rivest. April 1992. Available from: https://datatracker.ietf.org/doc/rfc1321/.

IETF. RFC 2046: *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Online. Edited by N. Freed, N. Borenstein. November 1996. Available from: https://datatracker.ietf.org/doc/rfc2046/.

IETF. RFC 2119: *Key words for use in RFCs to Indicate Requirement Levels*. Online. Edited by S. Bradner. Available from: https://datatracker.ietf.org/doc/rfc2119/.

IETF. RFC 2781: *UTF-16, an encoding of ISO 10646*. Online. Edited by P. Hoffman, F. Yergeau. February 2000. Available from: https://datatracker.ietf.org/doc/rfc2781/.

IETF. RFC 3339: *Date and Time on the Internet: Timestamps*. Online. Edited by G. Klyne, C. Newman. July 2002. Available from: https://datatracker.ietf.org/doc/rfc3339/.

IETF. RFC 3629 (STD 63): *UTF-8, a transformation format of ISO 10646*. Online. Edited by F. Yergeau. November 2003. Available from: https://datatracker.ietf.org/doc/rfc3629/.

IETF. RFC 3986 (STD 66): *Uniform Resource Identifier (URI): Generic Syntax*. Online. Edited by T. Berners-Lee, R. Fielding, L. Masinter. January 2005. Available from: https://datatracker.ietf.org/doc/rfc3986/.

IETF. RFC 3987: *Internationalized Resource Identifiers (IRIs).* Edited by M. Duerst and M. Suignard. IETF, January 2005. Available from: https://datatracker.ietf.org/doc/rfc3987/.

IETF. RFC 5234 (STD 68): *Augmented BNF for Syntax Specifications: ABNF*. Online. Edited by D. Crocker, P. Overell. January 2008. Available from: https://datatracker.ietf.org/doc/rfc5234/.

IETF. RFC 7405: *Case-Sensitive String Support in ABNF*. Online. Edited by P. Kyzivat. December 2014. Available from: https://datatracker.ietf.org/doc/rfc7405/.

IETF. RFC 9110 (STD 97): *HTTP Semantics*. Online. Edited by R. Fielding, M. Nottingham, J. Reschke. June 2022. Available from: https://datatracker.ietf.org/doc/rfc9110/.

THE UNICODE CONSORTIUM. *The Unicode Standard*. Online. Available from: https://www.unicode.org/standard/standard.html.

THE UNICODE CONSORTIUM. Unicode Standard Annex #9: *Unicode Bidirectional Algorithm*. Online. Edited by M. Goregaokar, R. Leroy. Available from: https://www.unicode.org/reports/tr9/.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub

Full document details: 25/30464096 DC

W3C. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1)*. Online. Edited by B. Bos, T. Çelik, I. Hickson, H. W. Lie. 7 June 2011. Available from: https://www.w3.org/TR/CSS2/.

W3C. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. Online. Edited by T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau. 16 August 2006. Available from: https://www.w3.org/TR/2006/REC-xml-20060816/.

W3C. *Namespaces in XML 1.0 (Third Edition)*. Online. Edited by T. Bray, D. Hollander, A. Layman, R. Tobin, H. S. Thompson. 8 December 2009. Available from: https://www.w3.org/TR/2009/REC-xml-names-20091208/.

W3C. *Scalable Vector Graphics (SVG) 1.1 (Second Edition)*. Online. Edited by E. Dahlström, P. Dengler, A. Grasso, C. Lilley, C. McCormack, D. Schepers, J. Watt. 16 August 2011. Available from: https://www.w3.org/TR/SVG11/.

W3C. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. Online. Edited by M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. F. Nielsen, A. Karmarkar, Y. Lafon. 27 April 2007. Available from: https://www.w3.org/TR/soap12/.

W3C. *XML Path Language (XPath) Version 1.0*. Online. Edited by J. Clark, S., DeRose. 16 November 1999. Available from: https://www.w3.org/TR/xpath-10/.

W3C. *XML Schema Part 1: Structures (Second Edition)*. Online. Edited by H. Thompson, D. Beech, M. Maloney, N. Mendelsohn. 28 October 2004. Available at: https://www.w3.org/TR/xmlschema-1/.

W3C. *XSL Transformations (XSLT) Version 1.0*. Online. Edited by J. Clark. 16 November 1999. Available from: https://www.w3.org/TR/xslt-10/.

CNS 7648: *Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times*. Ministry of Economic Affairs [Taiwan]: Bureau of Standards, Metrology and Inspection, 2015. Available from: https://www.cnsonline.com.tw/?node=detail&generalno=7648&locale=en_US.

INCITS 4-1986 (R2022) [formerly: ANSI X3.4-1986], *Information Systems - Coded Character Sets - 7-Bit Standard Code for Information Interchange (7-Bit ASCII)*. InterNational Committee for Information Technology Standards, 2022. Available from: https://webstore.ansi.org/standards/incits/incits1986r2022.

JIS X 0301: *Data elements and interchange formats —Information interchange — Representation of dates and times*. Japanese Standards Association Group, 2019. Available from: https://webdesk.jsa.or.jp/books/W11M0090/index/?bunsyo_id=JIS+X+0301%3A2002%2FAMENDMENT+1%3A2019.

HAR'EL, Zvi. *Gauss Formula for the Julian Date of Passover*. Online. Haifa: Technion – Israel Institute of Technology, 2005. Available from: http://harel.org.il/zvi/docs/gauss.pdf

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

MINISTRY OF EARTH SCIENCES [India]: Positional Astronomy Centre. *Indian Astronomical Ephemeris for the Year 2025*. Kolkata: Positional Astronomy Centre, 2024. Available from: https://packolkata.gov.in/iae.php. See also: https://packolkata.gov.in/indian-astronomical-ephemeris.php.

MINISTRY OF ISLAMIC AFFAIRS, ENDOWMENTS, DA'WAH AND GUIDANCE [Saudi Arabia].

REPUBLIC OF KOREA. Korean Law Enactment No. 4, 1961.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub

Full document details: 25/30464096 DC

SARGENT, Murray. Unicode Technical Note #28: *UnicodeMath, A Nearly Plain-Text Encoding of Mathematics*. Available from: https://www.unicode.org/notes/tn28/.

UNIVERSAL POSTAL UNION. *POST*CODE: Postal addressing systems*. Berne: UPU Publications, 2006, ISSN 1020-6019.

3. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at https://www.electropedia.org

3.1 General

3.1.1
package
ZIP archive conforming to the Open Packaging Conventions defined in ISO/IEC 29500-2

3.1.2
application
consumer or producer

3.1.3
consumer
piece of software or device that reads packages through a package implementer

Note 1 to entry: A consumer is often designed to consume packages only for a specific physical package format.

3.1.4
producer
piece of software or device that writes packages through a package implementer

Note 1 to entry: A producer is often designed to produce packages according to a particular physical package format specification.

3.1.5
Office Open XML document
rendition of a data stream formatted using WordprocessingML, SpreadsheetML, or PresentationML and its related MLs

Note 1 to entry: Such a document is represented as a package, as described in ISO/IEC 29500-2.

3.1.6
WordprocessingML
set of conventions for representing an Office Open XML document of category Wordprocessing

3.1.7
SpreadsheetML
set of conventions for representing an Office Open XML document of category Spreadsheet

3.1.8
PresentationML
set of conventions for representing an Office Open XML document of category Presentation

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

3.1.9
DrawingML
set of conventions for specifying the location and appearance of drawing elements in an Office Open XML document

3.1.10
document category
one of the three categories of Office Open XML documents (Wordprocessing, Spreadsheet, and Presentation)

Note 1 to entry: A document whose package-relationship item contains a relationship to…:

- …a Main Document part (§11.3.10) is a document of category Wordprocessing.
- …a Workbook part (§12.3.23) is a document of category Spreadsheet.
- …a Presentation part (§13.3.6) is a document of category Presentation.

Note 2 to entry: 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.

3.1.11
embedded package
package stored as the target of an Embedded Package relationship (§15.2.11) in an Office Open XML document

3.1.12
relationship
kind of connection between a source part and a target part in a package

Note 1 to entry: Relationships make the connections between parts directly discoverable without examining the content in the parts and without altering the parts themselves.

3.1.13
relationships part
part containing an XML representation of relationships

3.1.14
explicit relationship
relationship in which a resource is referenced from a source part's XML using the Id attribute of a Relationship tag

3.1.15
implicit relationship
relationship that is not explicit

3.1.16
behavior
external appearance or action

3.1.17
implementation-defined
unspecified behavior where each implementation is expected to document that behavior

Note 1 to entry: Documentation is intended to promote predictability and reproducibility within any given implementation.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

3.1.18
unspecified
application-dependent
<behavior> ISO/IEC 29500 makes no recommendations

Note 1 to entry: Extensions are properly added using the extensibility mechanisms described by ISO/IEC 29500, not by giving meaning to otherwise unspecified behavior.

3.1.19
byte
sequence of 8 bits treated as a unit

3.1.20
comment
note attached to content in a document

Note 1 to entry: 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.

3.1.21
content type
media type, subtype, and an optional set of parameters

Note 1 to entry: All instances of the term “content type” in this document are identical in meaning to the term “media type,” as defined in RFC 2046, that is preferred and used in other information technology standards.

3.1.22
extension
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

3.1.23
locale-specific
dependent on local conventions of nationality, culture, and language

3.1.24
ODBC
implementation of ISO/IEC 9075-3 or an SQL/CLI-based database connectivity API

EXAMPLE One broadly used SQL/CLI-based database connectivity API is the Open Database Connectivity (ODBC) API.

3.1.25
OLE
generalized abstraction of embedding and linking objects within a document

Note 1 to entry: OLE in the context of this document does not refer to any specific technology.

3.1.26
unique identifier
id
exclusive descriptor using one of a variety of different identification schemes

Note 1 to entry: In this document, the term does not imply use of the xsd:ID data type.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

3.2 WordprocessingML-specific

3.2.1
document setting
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

3.2.2
section
portion of a document in which certain page formatting options can be set

Note 1 to entry: A new section is created to change such properties as line numbering, number of columns, or headers and footers.

3.2.3
document building block
reusable element in a template

Note 1 to entry: Such elements include boilerplate text, cover pages, equations, footers, headers, tables, text boxes, and watermarks.

3.2.4
template
document that is a pattern for creating other documents

Note 1 to entry: A template can contain text, formatting, and graphics, among other things, such that documents based on it automatically have access to these elements.

3.2.5
master document
document that is the parent of one or more subdocuments

Note 1 to entry: A master document can be used to manage a multipart document, such as a book having several chapters. In such a 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.

3.2.6
subdocument
piece of a master document

Note 1 to entry: A chapter or appendix might be a subdocument in a book.

3.2.7
glossary document
additional WordprocessingML document story used to store reusable fragments of rich WordprocessingML content

Note 1 to entry: 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.

3.2.8
supplementary document storage location
part within a WordprocessingML document in which fragments of WordprocessingML content can be stored separate from the printed page

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

3.3 SpreadsheetML-specific

3.3.1
workbook
collection of worksheets

3.3.2
worksheet
two-dimensional grid of cells organized into rows and columns

3.3.3
cell
location at the intersection of a row and column, in which numeric or textual data or a formula is stored

Note 1 to entry: A cell can have a number of characteristics, such as numeric or text formatting, alignment, font, color, and border.

3.3.4
row
horizontal set of cells in a worksheet

Note 1 to entry: Each row has a numeric heading. Rows are numbered sequentially, starting at 1.

3.3.5
column
vertical set of cells in a worksheet

Note 1 to entry: Each column has an alphabetic heading. Columns are named sequentially, going from A–Z, then AA–AZ, BA–BZ, and so on.

3.3.6
cell reference
individual cell's designation using a combination of its column and row headings

EXAMPLE Cell references include A13, H19, and BX1200.

Note 1 to entry: 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 and does not 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.

3.3.7
formula
recipe for calculating a value

Note 1 to entry: Some formulas are predefined; others are user-defined.

3.3.8
function
predefined formula

Note 1 to entry: A function, such as AVERAGE, MAX, MIN, or SUM, takes one or more arguments on which it operates, producing a result. In the formula =SUM(B1:B4), there is one argument, B1:B4, which is the range of cells B1–B4, inclusive.

3.3.9
table

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

rectangular-shaped set of related rows and columns that can be sorted, filtered, and totaled as a group

Note 1 to entry: Rows in a table can be hidden by applying AutoFilters to one or more columns.

3.3.10
pivot table
kind of table used to manage and analyze related data that is stored elsewhere

3.3.11
AutoFilter
SpreadsheetML document state in which only certain rows are displayed, determined via filter criteria applied to the columns

3.3.12
chart
graphical representation of data

EXAMPLE Chart types include bar, column, line, and pie.

3.3.13
connection
means by which external data can be imported into a worksheet

Note 1 to entry: External data is data stored outside of a workbook, such as in a database or on a Web server.

3.3.14
MDX
multi-dimensional expression language

Note 1 to entry: MDX is passed to an OLAP provider. The method of interpreting this is defined by the server-side OLAP provider implementation.

3.3.15
OLAP
online analytical processing database which uses a multi-dimensional data model

3.4 PresentationML-specific

3.4.1
presentation
collection of slides intended to be viewed by an audience

3.4.2
handout
printed set of slides

Note 1 to entry: This is so named because the printed set is designed to be handed out to an audience for future reference.

3.4.3
slide
frame containing one or more pieces of text and/or images

3.4.4
slide layout
organization of elements on a slide

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

3.4.5
note
slide annotation, reminder, or piece of text intended for the presenter or the audience

3.5 DrawingML-specific

3.5.1
DiagramML
subset of DrawingML used for diagrams

3.5.2
diagram
picture or graphical representation displayed using a related set of color, data, layout, and style parts

EXAMPLE Diagram types include cycle, organization chart, pyramid, target, and Venn.

3.6 Shared

3.6.1
control
region of active content within an Office Open XML document

3.6.2
math zone
isolated region of text within which mathematical text is used and outside of which mathematical text is not used

3.6.3
mathematical text
text meant to convey mathematical meaning through OMML

3.6.4
OMML
Office Math Markup Language, a Shared ML of this document

3.6.5
instance of mathematical text
single continuous combination of mathematical text represented by a single `oMath` block and the OMML elements within that block

3.6.6
math paragraph
one or more instances of mathematical text that are in display mode

3.6.7
display mode
mathematical text in one or more `oMath` blocks contained in a display math zone (an `oMathPara` block)

3.6.8
oMath container
`oMath` block that is part of a display math zone but is not itself a math zone

3.6.9
display equation
display expression

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

display formula
display math
equation in display mode and thus part of a display math zone

3.6.10
inline equation
inline expression
inline formula
inline math
equation in an inline math zone

3.6.11
math alphanumerics
characters with specific math styles, as defined in the Unicode Standard

3.6.12
math accent
character specified as acceptable for use as an accent character by this document

3.6.13
linear format
implementation-defined, plain-text, 1-dimensional representation of mathematical text

3.6.14
built-down form
implementation-defined 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

3.6.15
professional form
built-up form
implementation-defined 2-D representation of mathematical text

3.6.16
build down
process of converting mathematical text from an implementation's professional form to an implementation's built-down form

3.6.17
build up
process of converting mathematical text from an implementation's built-down form to an implementation's professional form

3.6.18
limit
expression in a mathematical equation in reduced-size text, in a high position ("upper limit") or low position ("lower limit") relative to either an n-ary operator or text on the baseline of the expression

Note 1 to entry: See n-ary operator for examples of limits in both high and low positions relative to a summation sign operator.

3.6.19
n-ary operator
mathematical operator, typically a single operator character, optionally accompanied by expressions of a lower limit, an upper limit, or both

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

EXAMPLE The following example uses the summation sign (Unicode character U+2211), which has the official name “N-ARY SUMMATION”:

$$\sum_{j=1}^{n} a_j \equiv a_1 + a_2 + \cdots + a_n$$

4. Notational conventions

The following typographical conventions are used in this document:

- The first occurrence of a new term is written in italics. For example: "The text in this document is divided into normative and informative categories."
- The tag name of an XML element is written using a distinct style and typeface. For example: "The `bookmarkStart` and `bookmarkEnd` elements specify …"
- The name of an XML attribute is written using a distinct style and typeface. For example: "The `dropCap` attribute specifies …"
- The value of an XML attribute is written using a constant-width style. For example: "The attribute value of `auto` specifies …"
- The qualified or unqualified name of a simple type, complex type, or base datatype is written using a distinct style and typeface. For example: "The possible values for this attribute are defined by the `ST_HexColor` simple type."

Tag names in code fragments are generally expressed as prefixed names. The main exception is that tag names in SpreadsheetML examples are prefixless (§12, §18). Attribute names in code fragments are prefixless unless the name is declared to be in a specific namespace that is different to that of the parent element’s tag name, except in WordprocessingML examples in which all attribute names are prefixed.

References to an attribute of an element are given in XPath style notation. For example: "The value of `w:pgMar/@w:left` is …"

5. Organizational conventions

The primary table of contents does not list the full set of all subclauses in this document. Clauses and subclauses with many further subclauses have their own tables of contents which list all subordinate items.

Examples, which are informative and do not contain requirements, are shown using the lead-in “EXAMPLE” when separated from surrounding paragraphs and "For example, …" when inline with regular text.

The definitions of lists of items—such as attributes within elements, enumerated values within simple types, and other information whose inline presentation is improved by a grid layout—are presented in a format with the visual appearance of tables, but are not designated as tables. Sets of information referenced from other locations in this document are labeled as tables.

6. External data files

Normative requirements are prescribed in this document and the data files provided in electronic form, as noted in the Annexes, and all are required to fully implement ISO/IEC 29500. For example, default values of attributes and permissible contents of elements are specified in schemas only.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

7. Conformance

7.1 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 document
- The document is of category Wordprocessing, Spreadsheet, or Presentation, as defined in §3
- 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:
    - The Part may contain markup in the Markup Compatibility namespace as specified in ISO/IEC 29500-3
    - 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 schemas in Annex A.

This document 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

7.2 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.

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.

This document 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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

- 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 §7.3 for details.

7.3 Application descriptions

7.3.1 General

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

These application descriptions do not limit 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.

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, as specified by RFC 3986.

The application descriptions are defined in the following subclauses.

7.3.2 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.

In addition, applications that include a user interface are strongly recommended to support all accessibility features appropriate to that user interface.

7.3.3 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.

7.3.4 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.

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

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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. See References [2], [3], and [4].

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.

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

For the convenience of users of the description, it is recommended that creators of a description make a human- or machine-readable form of that description available at a URL corresponding to the description URI.

7.3.5 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.

It is recommended that implementers wishing to represent an application description within a document use the standard metadata mechanism for Office Open XML.

7.4 Interoperability guidelines

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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

The normative requirements in §7.1 imply that a conforming producer shall not write unescaped non-XML characters. As an implementation guideline, a conforming producer additionally should not write escaped non-XML characters. Doing so damages interoperability with existing XML-based standards such as SOAP. 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.

8. Overview

8.1 Content overview

This document contains predominantly the following three types of information:

1. Normative W3C XML Schema 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 part-relationship ZIP item for the source part. A package is an ordinary ZIP archive, which contains that package's content-type item, relationship items, and parts. (Packages are discussed further in ISO/IEC 29500-2.)

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

8.3 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.

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

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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

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

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

8.5 SpreadsheetML

This subclause introduces the overall form of a SpreadsheetML package, and identifies some of its main components.

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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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.

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

8.7.1 General

This subclause introduces the set of markup languages used across package categories.

The three markup languages described above define the structure of a package that is either a document (WordprocessingML), a spreadsheet (SpreadsheetML), or a presentation

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

(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.

8.7.2 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.

8.7.3 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.

8.7.4 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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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.

8.7.5 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

8.7.6 Bibliography

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

9. Packages

9.1 General

An Office Open XML document is stored as a package, whose format is defined by ISO/IEC 29500-2. The following subclauses contain 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.

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 subclause. Any requirement not mentioned here is inherited from the OPC specification.

9.2 Office Open XML's use of OPC

9.2.1 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.

9.2.2 Fragments

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

9.2.3 Physical packages

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

9.2.4 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.

9.2.5 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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

For example, a package has two parts that are 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
```

9.2.6 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.

9.2.7 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.

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.

9.2.8 Interleaving

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

9.3 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. Having multiple paths to a target can make access to that target more convenient.

Office Open XML imposes constraints on relationships, described in subsequent clauses of this document. 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. For 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. 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. Certain relationships shall be explicit. All other relationships are implicit. The syntax for specifying an implicit relationship varies based on the content model of the XML element which is the source of the relationship. Relationships that are required or permitted, and restrictions on those relationships are described in §10–15 of this document.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

EXAMPLE 1

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

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

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

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

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

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

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

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

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

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

END OF EXAMPLE

EXAMPLE 2

Figure 1 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 implicitly corresponds to the value of the target (for instance, "footnoteReference" in the source indicates the relationship type "footnotes" in the relationship item).

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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. These relationships are illustrated in **Figure 2** and **Figure 3**.

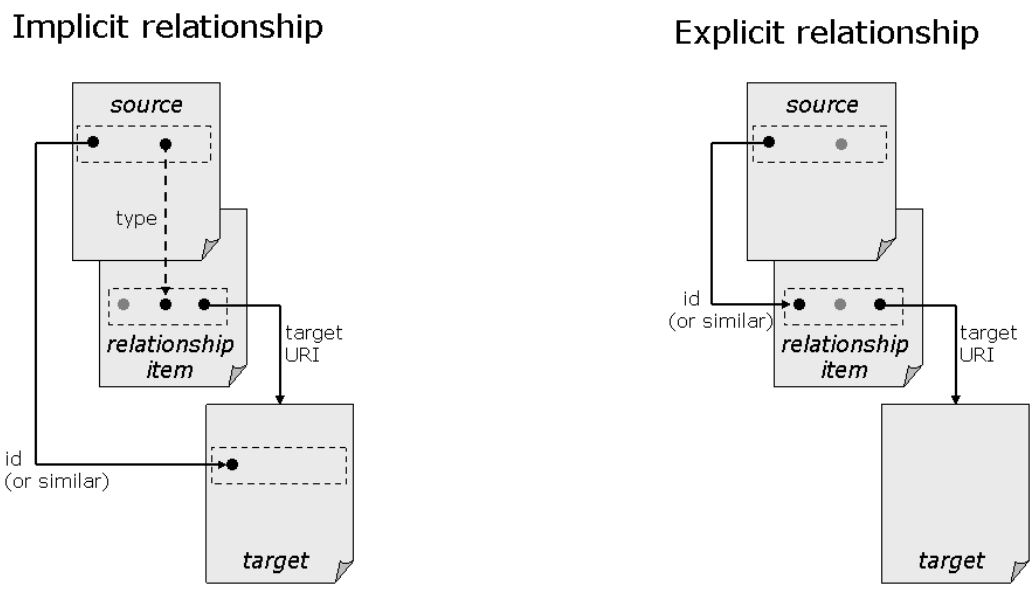


Figure 1 — Implicit and explicit relationships

END OF EXAMPLE

EXAMPLE 3

Figure 2 below shows the implicit relationship for the footnote example described earlier.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

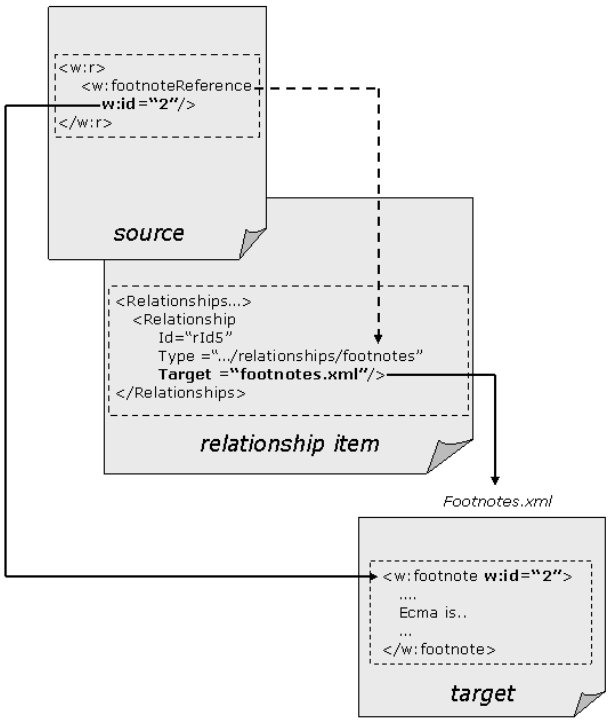


Figure 2 — Implicit relationship

END OF EXAMPLE

EXAMPLE 4

Figure 3 below shows an explicit relationship.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

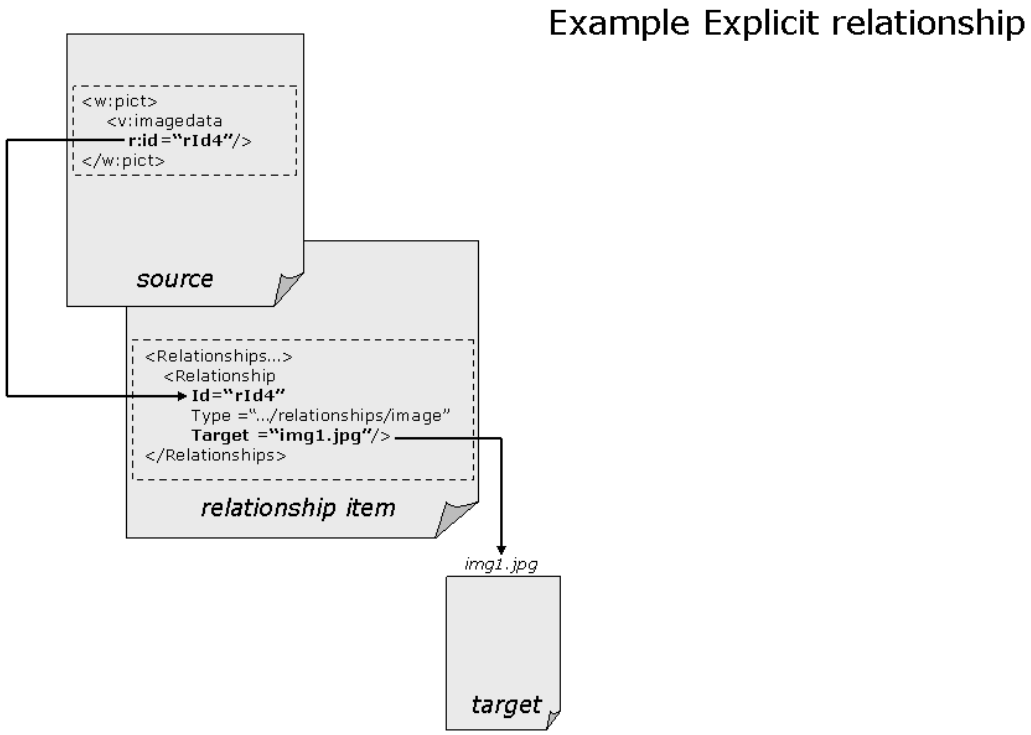


Figure 3 — Explicit relationship

END OF EXAMPLE

10. Markup compatibility and extensibility

10.1 General

Office Open XML documents are designed to allow for innovation by extending their capabilities using the Application-Defined Extension Elements specified by this document or the Markup Compatibility and Extensibility features specified by ISO/IEC 29500-3.

10.2 Validity and MCE

This clause describes MCE processing for validation of Office Open XML documents. The OPC parts shown below shall be valid after the removal of any extensions by an MCE processor. Non-XML-based parts and user-defined parts are omitted.

- Comments part (§11.3.2)
- Document Settings part (§11.3.3)
- Endnotes part (§11.3.4)
- Font table part (§11.3.5)
- Footer part (§11.3.6)
- Footnotes part (§11.3.7)
- Glossary Document part (§11.3.8)
- Header part (§11.3.9)
- Main Document part (§11.3.10)

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

- Numbering Definitions part (§11.3.11)
- Style Definitions part (§11.3.12)
- Web Settings part (§11.3.13)
- Mail Merge Recipient Data part (§11.3.14)
- Calculation Chain part (§12.3.1)
- Chartsheet part (§12.3.2)
- Comments part (§12.3.3)
- Connections part (§12.3.4)
- Custom XML Mappings part (§12.3.6)
- Dialogsheet part (§12.3.7)
- Drawing part (§12.3.8)
- External Workbook References part (§12.3.9)
- Metadata part (§12.3.10)
- Pivot table part (§12.3.11)
- Pivot Table Cache Definition part (§12.3.12)
- Pivot Table Cache Records part (§12.3.13)
- Query Table part (§12.3.14)
- Shared String Table part (§12.3.15)
- Shared Workbook Revision Headers part (§12.3.16)
- Shared Workbook Revision Log part (§12.3.17)
- Shared Workbook User Data part (§12.3.18)
- Single Cell Table Definitions part (§12.3.19)
- Styles part (§12.3.20)
- Table Definition part (§12.3.21)
- Volatile Dependencies part (§12.3.22)
- Workbook part (§12.3.23)
- Worksheet part (§12.3.24)
- Comment Authors (§13.3.1)
- Comments part (§13.3.2)
- Handout Master part (§13.3.3)
- Notes Master part (§13.3.4)
- Notes Slide part (§13.3.5)
- Presentation part (§13.3.6)
- Presentation Properties part (§13.3.7)
- Slide part (§13.3.8)
- Slide Layout part (§13.3.9)
- Slide Master part (§13.3.10)
- Slide Synchronization Data part (§13.3.11)
- User Defined Tags part (§13.3.12)
- View Properties part (§13.3.13)
- Chart part (§14.2.1)
- Chart Drawing part (§14.2.2)
- Diagram Colors part (§14.2.3)
- Diagram Data part (§14.2.4)
- Diagram Layout Definition part (§14.2.5)
- Diagram Style part (§14.2.6)
- Table Styles part (§14.2.7)

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

- Theme part (§14.2.8)
- Theme Override part (§14.2.9)
- Additional Characteristics part (§15.2.1)
- Bibliography part (§15.2.3)
- Custom XML Data Storage Properties part (§15.2.6)
- Custom File Properties part (§15.2.12.3)
- Extended File Properties part (§15.2.12.4)

The markup configuration for MCE processing, as specified in ISO/IEC 29500-3, shall be a set containing the expanded names of the `ext` element (§18.2.7), the `ext` element (§19.2.1.11), the `ext` element (§20.1.2.2.14), and the `ext` element (§21.2.2.62). The application configuration for MCE processing shall be the empty set. The file containing the schema for each part is shown in §16.

Since the application configuration is the empty set, the MCE processor selects no child `Choice` elements of `AlternateContent` elements. It always selects `Fallback` child elements.

The validity of parts defined in OPC—the Digital Signature Origin part, the Digital Signature XML Signature part, and the Core File Properties part—is specified in ISO/IEC 29500-2.

MCE processing may be applied to other user-defined XML OPC parts.

10.3 Applications and MCE

10.3.1 General

This clause describes MCE processing for building applications. The application configuration used during MCE processing for building applications varies depending on the namespaces supported by the application.

An application may use an MCE processor but is not required to do so. However, an application shall handle MCE constructs in a manner consistent with ISO/IEC 29500-3.

For example, editors are applications that are consumers and producers as the same time. When such editors receive documents containing `AlternateContent` blocks, they might preserve all child `Choice` and `Fallback` elements, rather than discarding unselected ones for round-tripping without loss of information. Nevertheless, for rendering, such editors shall use the `Choice` and `Fallback` elements that would be selected by an MCE processor.

10.3.2 Application implementing no extensions

The application configuration for a consumer that implements this document and no extensions shall be the empty set.

The MCE processor removes all elements in ignorable namespaces, which are specified by the `Ignorable` attribute as defined in ISO/IEC 29500-3. Thus, consumers do not receive elements in ignorable namespaces. The MCE processor does not choose any `Choice` elements but instead chooses `Fallback` elements.

10.3.3 Examples

EXAMPLE 1

Consider an application that implements the extensions described in [MS-DOCX] version 21.1 and no other extensions.

The application configuration is a set containing the following namespaces:

http://schemas.microsoft.com/office/word/2010/wordml

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

http://schemas.microsoft.com/office/word/2012/wordml
http://schemas.microsoft.com/office/word/2015/wordml/symex
http://schemas.microsoft.com/office/word/2016/wordml/cid
http://schemas.microsoft.com/office/word/2018/wordml
http://schemas.microsoft.com/office/word/2018/wordml/cex
http://schemas.microsoft.com/office/word/2020/wordml/sdtdatahash
http://schemas.microsoft.com/office/word/2023/wordml/word16du

The MCE processor as specified by ISO/IEC 29500-3 does not remove elements in these namespaces even when they are ignorable namespaces. Thus, consumers receive such elements. The MCE processor chooses `Choice` elements having such namespaces and no other namespaces.

END EXAMPLE

EXAMPLE 2

Consider an application that implements the extensions described in [MS-DOCX] version 3.0 and no other extensions.

The application configuration is a set containing the following namespaces:

http://schemas.microsoft.com/office/word/2010/wordml
http://schemas.microsoft.com/office/word/2012/wordml

The MCE processor as specified by ISO/IEC 29500-3 does not remove elements in these namespaces but removes elements of the namespace http://schemas.microsoft.com/office/word/2018/wordml. Thus, consumers receive elements of the namespace http://schemas.microsoft.com/office/word/2010/wordml or http://schemas.microsoft.com/office/word/2012/wordml but do not receive elements of http://schemas.microsoft.com/office/word/2018/wordml.

END EXAMPLE

EXAMPLE 3

Consider a producer that implements [MS-ODRAWXML] version 27.0. Documents created by this producer may contain the following `AlternateContent` element:

```
<mc:AlternateContent
 xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
  <mc:Choice Requires="c14"
   xmlns:c14="http://schemas.microsoft.com/office/drawing/2007/8/2/chart">
    <c14:style val="102"/>
  </mc:Choice>
  <mc:Fallback><c:style val="2"/></mc:Fallback>
</mc:AlternateContent>
```

Consider a consumer that is built using an MCE processor. If this consumer does not implement [MS-ODRAWXML] version 27.0, the application configuration for this consumer does not contain the namespace http://schemas.microsoft.com/office/drawing/2007/8/2/chart and the consumer receives the `c:style` element in the `mc:Fallback` element. If this consumer does implement [MS-ODRAWXML] version 27.0, the application configuration includes this namespace and the consumer receives the `c14:style` element in the `mc:Choice` element.

END EXAMPLE

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

11. WordprocessingML

11.1 General

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.

11.2 Package structure

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

- One Main Document part (§11.3.10)

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

- Digital Signature Origin (§15.2.7)
- File Property parts (§15.2.12)
- Thumbnail (§15.2.16)

The required and optional relationships between parts are defined in §11.3 and its subordinate clauses.

EXAMPLE 1

The following package represents the minimal conformant WordprocessingML package as defined by this document:

First, the content type for relationship parts and the Main Document part (the only required part) is 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) is defined (physically located at /_rels/.rels in the package):

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

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

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

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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

END OF EXAMPLE

EXAMPLE 2

Consider a WordprocessingML document that is an early draft of this document. 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                 Extended File Properties part
/docProps/core.xml                Core File Properties part
/word/document.xml                Main Document part
/word/_rels/document.xml.rels     Part-relationship item
/word/comments.xml                Comment part
/word/endnotes.xml                Endnotes part
/word/fontTable.xml               Font Table part
/word/footer1.xml                 Footer parts
/word/footer2.xml
/word/footer3.xml
/word/footer4.xml
/word/footnotes.xml               Footnotes part
/word/header1.xml                 Header parts
/word/header2.xml
/word/header3.xml
/word/header4.xml
/word/header5.xml
/word/header6.xml
/word/numbering.xml               Numbering Definitions part
/word/settings.xml                Document Settings part
/word/styles.xml                  Style Definitions part
/word/theme/theme1.xml            Theme part
```

The package-relationship item contains the following:

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

END OF EXAMPLE

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

11.3 Parts

11.3.1 Alternative Format Import part

Content Type:	Any text-based content, support for which is implementation-defined. 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
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.

This document 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.

The Alternative Format Import machinery provides a one-time conversion facility. A conforming 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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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 OF 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 this document.

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

- HTML - application/text/html
- TEXT - application/text/plain (UTF-16, as specified by RFC 2781)

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.

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

For 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>
```

The root element for a Comment part shall be `comments`.

For example:

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

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

For 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>
```

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 OF 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 this document:

- Alternative Format Import (§11.3.1)

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

- 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 this document.

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.

For 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>
```

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>
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

```
    <w:rsids>
    …
    </w:rsids>
    …
  </w:settings>
```

END OF 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 this document:

- 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 this document.

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.

For 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>
```

The root element for an Endnotes part shall be `endnotes`.

For example:

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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

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

For 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>
```

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>
```

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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

END OF 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 this document:

- 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 this document.

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.

For 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:

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC

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

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 OF 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 this document:

- Fonts (§15.2.13)

A Font Table part shall not have any implicit or explicit relationships to any other part defined by this document.

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 part-relationship item for the Main Document (§11.3.10) part, or the Glossary Document (§11.3.8) part.

For example, the Main Document part-relationship item contains one relationship, for the odd footer part, which is stored as the ZIP item footer3.xml:

© ISO/IEC 2025 – All rights reserved

Documents
Document Sharing Hub
Full document details: 25/30464096 DC