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

**Part 2:
Open Packaging Conventions**

iTeh STANDARD PREVIEW
(standards.iteh.ai)
*Technologies de l'information — Description des documents et
langages de traitement — Formats de fichier "Office Open XML" —
Partie 2: Conventions de paquetage ouvert*

[ISO/IEC 29500-2:2011](https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 29500-2:2011](https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

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

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

Published in Switzerland

Table of Contents

Foreword.....	vii
Introduction	viii
1. Scope	1
2. Conformance	2
3. Normative References.....	3
4. Terms and Definitions	5
5. Notational Conventions	8
5.1 Document Conventions.....	8
5.2 Diagram Notes.....	8
6. Acronyms and Abbreviations.....	10
7. General Description	11
8. Overview	12
9. Package Model	13
9.1 Parts.....	13
9.1.1 Part Names.....	13
9.1.2 Content Types	16
9.1.3 Growth Hint.....	17
9.1.4 XML Usage.....	17
9.2 Part Addressing	18
9.2.1 Relative References.....	18
9.2.2 Fragments.....	19
9.3 Relationships	19
9.3.1 Relationships Part.....	19
9.3.2 Relationship Markup	20
9.3.3 Representing Relationships.....	22
9.3.4 Support for Versioning and Extensibility.....	25
10. Physical Package	26
10.1 Physical Mapping Guidelines.....	26
10.1.1 Mapped Components.....	27
10.1.2 Mapping Content Types	27
10.1.3 Mapping Part Names to Physical Package Item Names.....	32
10.1.4 Interleaving	34
10.2 Mapping to a ZIP Archive	36
10.2.1 Mapping Part Data	36
10.2.2 ZIP Item Names	36
10.2.3 Mapping Part Names to ZIP Item Names.....	37
10.2.4 Mapping ZIP Item Names to Part Names	37
10.2.5 ZIP Package Limitations.....	37
10.2.6 Mapping Part Content Type	38

ITeH STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 29500-2:2011](https://standards.iteh.ai/catalog/standards/sist/3bef176a-flab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

<https://standards.iteh.ai/catalog/standards/sist/3bef176a-flab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

10.2.7	Mapping the Growth Hint	38
10.2.8	Late Detection of ZIP Items Unfit for Streaming Consumption	39
10.2.9	ZIP Format Clarifications for Packages	39
11.	Core Properties	40
11.1	Core Properties Part	41
11.2	Location of Core Properties Part	43
11.3	Support for Versioning and Extensibility	43
11.4	Schema Restrictions for Core Properties	43
12.	Thumbnails	45
12.1	Thumbnail Parts	45
13.	Digital Signatures	46
13.1	Choosing Content to Sign	46
13.2	Digital Signature Parts	46
13.2.1	Digital Signature Origin Part	47
13.2.2	Digital Signature XML Signature Part	47
13.2.3	Digital Signature Certificate Part	48
13.2.4	Digital Signature Markup	48
13.3	Digital Signature Example	58
13.4	Generating Signatures	60
13.5	Validating Signatures	61
13.5.1	Signature Validation and Streaming Consumption	62
13.6	Support for Versioning and Extensibility	62
13.6.1	Using Relationship Types	62
13.6.2	Markup Compatibility Namespace for Package Digital Signatures	62
Annex A.	(normative) Resolving Unicode Strings to Part Names	65
A.1	Creating an IRI from a Unicode String	65
A.2	Creating a URI from an IRI	65
A.3	Resolving a Relative Reference to a Part Name	66
A.4	String Conversion Examples	66
Annex B.	(normative) Pack URI	67
B.1	Pack URI Scheme	67
B.2	Resolving a Pack URI to a Resource	69
B.3	Composing a Pack URI	69
B.4	Equivalence	70
Annex C.	(normative) ZIP Appnote.txt Clarifications	71
C.1	Archive File Header Consistency	71
C.2	Table Key	71
Annex D.	(normative) Schemas - W3C XML Schema	82
D.1	Content Types Stream	82
D.2	Core Properties Part	83
D.3	Digital Signature XML Signature Markup	84
D.4	Relationships Part	85
Annex E.	(informative) Schemas - RELAX NG	86

E.1	Content Types Stream	86
E.2	Core Properties Part	87
E.3	Digital Signature XML Signature Markup	87
E.4	Relationships Part.....	88
E.5	Additional Resources.....	89
E.5.1	XML.....	89
E.5.2	XML Digital Signature Core.....	89
Annex F. (normative) Standard Namespaces and Content Types.....		90
Annex G. (informative) Physical Model Design Considerations		92
G.1	Access Styles.....	93
G.1.1	Direct Access Consumption.....	93
G.1.2	Streaming Consumption.....	93
G.1.3	Streaming Creation	93
G.1.4	Simultaneous Creation and Consumption	93
G.2	Layout Styles.....	93
G.2.1	Simple Ordering.....	93
G.2.2	Interleaved Ordering.....	94
G.3	Communication Styles.....	94
G.3.1	Sequential Delivery	94
G.3.2	Random Access.....	94
Annex H. (informative) Guidelines for Meeting Conformance.....		95
H.1	Package Model	95
H.2	Physical Packages	103
H.3	ZIP Physical Mapping.....	108
H.4	Core Properties.....	112
H.5	Thumbnail.....	114
H.6	Digital Signatures.....	114
H.7	Pack URI.....	125
Annex I. (informative) Differences Between ISO/IEC 29500 and ECMA-376:2006		127
I.1	XML Elements.....	127
I.2	XML Attributes.....	127
I.3	XML Enumeration Values	127
I.4	XML Simple Types.....	127
Annex J. (informative) Index.....		128

ITeH STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 29500-2:2011

[https://standards.iteh.ai/catalog/standards/sist/3bef176a-44ab-4076-89ae-](https://standards.iteh.ai/catalog/standards/sist/3bef176a-44ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

[62204ec75aa7/iso-iec-29500-2-2011](https://standards.iteh.ai/catalog/standards/sist/3bef176a-44ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 29500-2:2011

<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

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

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

ISO/IEC 29500-2 was prepared by ISO/IEC JTC 1, Information technology, Subcommittee SC 34, Document description and processing languages.

This second edition cancels and replaces the first edition (ISO/IEC 29500-2:2008), which has been technically revised by incorporation of the Technical Corrigendum ISO/IEC 29500-2:2008/Cor.1:2010.

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

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

Annexes A, B, C, D, and F form a normative part of this Part of ISO/IEC 29500. Annexes E, G, H, I, and J are for information only.

This Part of ISO/IEC 29500 includes two annexes (Annex D and Annex E) that refer to data files provided in electronic form.

Introduction

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

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

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

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 29500-2:2011](https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011)

<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

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

Part 2:

Open Packaging Conventions

1. Scope

This Part of ISO/IEC 29500 specifies a set of conventions that are used by Office Open XML documents to define the structure and functionality of a *package* in terms of a package model and a physical model.

The *package model* is a package abstraction that holds a collection of *parts*. The parts are composed, processed, and persisted according to a set of rules. Parts can have relationships to other parts or external resources, and the package as a whole can have relationships to parts it contains or to external resources. The package model specifies how the parts of a package are named and related. Parts have content types and are uniquely identified using the well-defined naming rules provided in this Part of ISO/IEC 29500.

The *physical mapping* defines the mapping of the components of the package model to the features of a specific physical format, namely a ZIP archive.

This Part of ISO/IEC 29500 also describes certain features that might be supported in a package, including *core properties* for package metadata, a *thumbnail* for graphical representation of a package, and *digital signatures* of package contents.

Because this Part of ISO/IEC 29500 might evolve, packages are designed to accommodate extensions and to support compatibility goals in a limited way. The versioning and extensibility mechanisms described in Part 3 support compatibility between software systems based on different versions of this Part of ISO/IEC 29500 while allowing package creators to make use of new or proprietary features.

This Part of ISO/IEC 29500 specifies requirements for documents, producers, and consumers. Conformance requirements are identified throughout the text of this Part of ISO/IEC 29500. A formal conformance statement is given in §2. An informative summary of requirements relevant to particular classes of developers is given in Annex H.

2. Conformance

Each conformance requirement is given a unique ID comprised of a letter (M – MANDATORY; S – SHOULD; O – OPTIONAL), an identifier for the topic to which it relates, and a unique ID within that topic. (Producers and consumers might use these IDs to report error conditions.) Mandatory requirements are those stated with the normative terms "shall," "shall not," or any of their normative equivalents. Should items are those stated with the normative terms "should," "should not," or any of their normative equivalents. Optional requirements are those stated with the normative terms "can," "cannot," "might," "might not," or any of their normative equivalents.

[*Example*: Package implementers shall not map logical item name(s) mapped to the Content Types stream in a ZIP archive to a part name. [M3.11] *end example*]

Each Part of this multi-part standard has its own conformance clause, as appropriate. The term *conformance class* is used to disambiguate conformance within different Parts of this multi-part standard. This Part of ISO/IEC 29500 has only one conformance class, *OPC* (that is, Open Packaging Conventions).

A document is of conformance class OPC if it obeys all syntactic constraints specified in this Part of ISO/IEC 29500.

OPC conformance is purely syntactic.

ISO/IEC 29500-2:2011
<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

3. Normative References

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

American National Standards Institute, *Coded Character Set — 7-bit American Standard Code for Information Interchange*, ANSI X3.4, 1986.

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

ISO/IEC 9594-8 | ITU-T Rec. X.509, *Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*.

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

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

Dublin Core Element Set v1.1. <http://purl.org/dc/elements/1.1/>

Dublin Core Terms Namespace. <http://purl.org/dc/terms/>

Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, 04 February 2004.

Namespaces in XML 1.1, W3C Recommendation, 4 February 2004.

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

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

RFC 3987 *Internationalized Resource Identifiers (IRIs)*, The Internet Society, Duerst, M. and M. Suignard, 2005, <http://www.ietf.org/rfc/rfc3987.txt>.

RFC 4234 *Augmented BNF for Syntax Specifications: ABNF*, The Internet Society, Crocker, D., (editor), 2005, <http://www.ietf.org/rfc/rfc4234.txt>.

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

W3C NOTE 19980827, *Date and Time Formats*, Wicksteed, Charles, and Misha Wolf, 1997, <http://www.w3.org/TR/1998/NOTE-datetime-19980827>.

XML, Tim Bray, Jean Paoli, Eve Maler, C. M. Sperberg-McQueen, and François Yergeau (editors). *Extensible Markup Language (XML) 1.0, Fourth Edition*. World Wide Web Consortium. 2006.

<http://www.w3.org/TR/2006/REC-xml-20060816/>. [Implementers should be aware that a further correction of the normative reference to XML to refer to the 5th Edition will be necessary when the related Reference Specifications to which this International Standard also makes normative reference and which also depend upon XML, such as XSLT, XML Namespaces and XML Base, are all aligned with the 5th Edition.]

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

XML Base, W3C Recommendation, 27 June 2001.

XML Path Language (XPath), Version 1.0, W3C Recommendation, 16 November 1999.

XML Schema Part 1: Structures, W3C Recommendation, 28 October 2004.

XML Schema Part 2: Datatypes, W3C Recommendation, 28 October 2004.

XML-Signature Syntax and Processing, W3C Recommendation, 12 February 2002.

.ZIP File Format Specification from PKWARE, Inc., version 6.2.0 (2004), as specified in http://www.pkware.com/documents/APPNOTE/APPNOTE_6.2.0.txt. [Note: The supported compression algorithm is inferred from tables C-3 and C-4 in Annex C. *end note*]

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 29500-2:2011
<https://standards.iteh.ai/catalog/standards/sist/3bef176a-f4ab-4076-89ae-62204ec75aa7/iso-iec-29500-2-2011>

4. Terms and Definitions

For the purposes of this document, the following terms and definitions apply. Other terms are defined where they appear in italic typeface. Terms explicitly defined in this Part of ISO/IEC 29500 are not to be presumed to refer implicitly to similar terms defined elsewhere.

The terms *base URI* and *relative reference* are used in accordance with RFC 3986.

access style — The style in which local access or networked access is conducted. The access styles are as follows: streaming creation, streaming consumption, simultaneous creation and consumption, and direct access consumption.

behavior — External appearance or action.

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

behavior, unspecified — Behavior where this Open Packaging specification imposes no requirements.

communication style — The style in which package contents are delivered by a producer or received by a consumer. Communication styles include random access and sequential delivery.

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

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

Content Types stream — A specially-named stream that defines mappings from part names to content types. The content types stream is not itself a part, and is not URI addressable.

device — A piece of hardware, such as a personal computer, printer, or scanner, that performs a single function or set of functions.

format consumer — A consumer that consumes packages conforming to a format designer's specification.

format designer — The author of a particular file format specification built on this Open Packaging Conventions specification.

format producer — A producer that produces packages conforming to a format designer's specification.

growth hint — A suggested number of bytes to reserve for a part to grow in-place.

interleaved ordering — The layout style of a physical package where parts are broken into pieces and “mixed-in” with pieces from other parts. When delivered, interleaved packages can help improve the performance of the consumer processing the package.

layout style — The style in which the collection of parts in a physical package is laid out: either simple ordering or interleaved ordering.

local access — The access architecture in which a pipe carries data directly from a producer to a consumer on a single device.

logical item name — An abstraction that allows package implementers to manipulate physical data items consistently regardless of whether those data items can be mapped to parts or not or whether the package is laid out with simple ordering or interleaved ordering.

networked access — The access architecture in which a consumer and the producer communicate over a protocol, such as across a process boundary, or between a server and a desktop computer.

pack URI — A URI scheme that allows URIs to be used as a uniform mechanism for addressing parts within a package. Pack URIs are used as Base URIs for resolving relative references among parts in a package.

package — A logical entity that holds a collection of parts.

package implementer — Software that implements the physical input-output operations to a package according to the requirements and recommendations of this Open Packaging specification. A package implementer is used by a producer or consumer to interact with a physical package. A package implementer can be either a stand-alone API or can be an integrated component of a producer, consumer application, or device.

package model — A package abstraction that holds a collection of parts.

package relationship — A relationship whose target is a part and whose source is the package as a whole. Package relationships are found in the package relationships part named “/_rels/.rels”.

part — A stream of bytes with a MIME content type and associated common properties. Typically corresponds to a file [*Example*: on a file system *end example*], a stream [*Example*: in a compound file *end example*], or a resource [*Example*: in an HTTP URI *end example*].

part name — The path component of a pack URI. Part names are used to refer to a part in the context of a package, typically as part of a URI.

physical model — A description of the capabilities of a particular physical format.

physical package format — A specific file format, or other persistence or transport mechanism, that can represent all of the capabilities of a package.

piece — A portion of a part. Pieces of different parts can be interleaved together. The individual pieces are named using a unique mapping from the part name. Piece name grammar is not equivalent to the part name grammar. Pieces are not addressable in the package model.

pipe — A communication mechanism that carries data from the producer to the consumer.

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

random access — A style of communication between the producer and the consumer of the package. Random access allows the consumer to reference and obtain data from anywhere within a package.

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

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

sequential delivery — A communication style in which all of the physical bits in the package are delivered in the order they appear in the package.

signature policy — A format-defined policy that specifies what configuration of parts and relationships shall or might be included in a signature for that format and what additional behaviors that producers and consumers of that format shall follow when applying or verifying signatures following that format's signature policy.

simple ordering — A defined ordering for laying out the parts in a package in which all the bits comprising each part are stored contiguously.

simultaneous creation and consumption — A style of access between a producer and a consumer in highly pipelined environments where streaming creation and streaming consumption occur simultaneously.

stream — A linearly ordered sequence of bytes.

streaming consumption — An access style in which parts of a physical package can be processed by a consumer before all of the bits of the package have been delivered through the pipe.

streaming creation — A production style in which a producer dynamically adds parts to a package after other parts have been added without modifying those parts.

thumbnail — A small image that is a graphical representation of a part or the package as a whole.

well-known part — A part with a well-known relationship, which enables the part to be found without knowing the location of other parts.

XSD — W3C XML Schema

ZIP archive — A ZIP file as defined in the ZIP file format specification. A ZIP archive contains ZIP items.

ZIP item — A ZIP item is an atomic set of data in a ZIP archive that becomes a file when the archive is uncompressed. When a user unzips a ZIP-based package, the user sees an organized set of files and folders.