FINAL
DRAFT

INTERNATIONAL
STANDARD

**ISO/IEC
FDIS
29500-2**

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

Secretariat: JISC

Voting begins on:
2021-05-10

Voting terminates on:
2021-07-05

Document description and processing languages — Office Open XML file formats —

Part 2: Open packaging conventions

Description des documents et langages de traitement — Formats de fichier "Office Open XML" —
Partie 2: Conventions de paquetage ouvert

Reference number
ISO/IEC FDIS 29500-2:2021(E)

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

Contents

Page

Foreword ..... **v**

Introduction ..... **vii**

1 Scope **1**

2 Normative references **1**

3 Terms and definitions **2**
3.1 Basics 2
3.2 Abstract package model 3
3.3 Physical package model 4
3.4 Digital signature and thumbnail 5
3.5 Implementations 5
3.6 Core properties 5

4 Conformance **5**

5 Overview **5**

6 Abstract package model **6**
6.1 General 6
6.2 Parts 6
6.2.1 General 6
6.2.2 Part names 6
6.2.3 Media types 8
6.2.4 Growth hint 8
6.2.5 XML usage 8
6.3 Part addressing 8
6.3.1 General 8
6.3.2 Pack scheme 9
6.3.3 Resolving a pack IRI to a resource 10
6.3.4 Composing a pack IRI 10
6.3.5 Equivalence 11
6.4 Resolving relative references 11
6.4.1 General 11
6.4.2 Base IRIs 11
6.4.3 Examples 12
6.5 Relationships 14
6.5.1 General 14
6.5.2 Relationships part 14
6.5.3 Relationship markup 15
6.5.4 Examples 17

7 Physical package model **19**
7.1 General 19
7.2 Physical mapping guidelines 19
7.2.1 Using features of physical formats 19
7.2.2 Mapped components 20
7.2.3 Mapping media types to parts 20
7.2.4 Interleaving 23
7.2.5 Mapping part names to physical package item names 23
7.3 Mapping to a ZIP file 25
7.3.1 General 25
7.3.2 Mapping part data 25
7.3.3 ZIP item names 26
7.3.4 Mapping logical item names to ZIP item names 26
7.3.5 Mapping ZIP item names to logical item names 26
7.3.6 ZIP package limitations 26
7.3.7 Mapping the Media Types stream 27

7.3.8 Mapping the growth hint ... 27
8 Core properties ... 27
8.1 General ... 27
8.2 Core Properties part ... 28
8.3 Core properties markup ... 28
8.3.1 General ... 28
8.3.2 Support for versioning and extensibility ... 29
8.3.3 `coreProperties` element ... 29
8.3.4 Core property elements ... 29
9 Thumbnails ... 32
10 Digital signatures ... 32
10.1 General ... 32
10.2 Overview of OPC-specific restrictions and extensions to "XML-Signature Syntax and Processing" ... 32
10.3 Choosing content to sign ... 32
10.4 Digital signature parts ... 33
10.4.1 General ... 33
10.4.2 Digital Signature Origin part ... 33
10.4.3 Digital Signature XML Signature part ... 33
10.4.4 Digital Signature Certificate part ... 34
10.5 Digital signature markup ... 34
10.5.1 General ... 34
10.5.2 Support for versioning and extensibility ... 34
10.5.3 `Signature` element ... 34
10.5.4 `SignedInfo` element ... 35
10.5.5 `CanonicalizationMethod` element ... 35
10.5.6 `SignatureMethod` element ... 35
10.5.7 `Reference` element ... 35
10.5.8 `Transform` element ... 36
10.5.9 `RelationshipReference` element ... 37
10.5.10 `RelationshipsGroupReference` element ... 37
10.5.11 `DigestMethod` element ... 37
10.5.12 `Object` element ... 38
10.5.13 `Manifest` element ... 38
10.5.14 `SignatureProperty` element ... 38
10.5.15 `SignatureTime` element ... 38
10.5.16 `Format` element ... 38
10.5.17 `Value` element ... 39
10.5.18 `XPath` element ... 39
10.6 Relationships transform algorithm ... 39
10.7 Digital signature example ... 40
10.8 Generating signatures ... 41
10.9 Validating signatures ... 43
Annex A (informative) **Preprocessing for generating relative references ... 45**
Annex B (normative) **Constraints and clarifications on the use of ZIP features ... 46**
Annex C (normative) **Schemas - W3C XML ... 55**
Annex D (informative) **Schemas - RELAX NG ... 56**
Annex E (normative) **Standard namespaces and media types ... 57**
Annex F (informative) **Physical package model design considerations ... 58**
Annex G (informative) **Differences between ISO/IEC 29500-2 and ECMA-376:2006 ... 62**
Annex H (informative) **Package example ... 63**
Bibliography ... 65

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).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

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 fourth edition cancels and replaces the third edition (ISO/IEC 29500-2:2012), which has been technically revised.

The main changes compared to the previous edition are as follows:

— Where appropriate, normative references have been updated to use undated or more recent versions of other standards.

— Clause 3 (Terms and definitions) has been revised by removing terms not used by any normative clauses and then reorganizing terms into groups.

— The subclause for diagram notes (5.1 in the preceding editions) has been removed, since core properties are now defined by prose and schemas rather than by diagrams.

— The clause for acronyms and abbreviations (Clause 6 in the preceding editions) has been removed, since it does not make sense for an ISO/IEC standard to define "ISO" and "IEC".

— Clause 6 (Abstract package model, Clause 8 in the previous edition) has been completely rewritten. In particular, (1) pack IRIs have been defined in this clause rather than in an annex, (2) a new subclause, "Resolving relative references", has been added; (3) part Relationships parts and package Relationships parts have been distinguished; (4) base IRIs have been clearly defined; and (5) handling of non-ASCII characters in part names has been clarified on the basis of RFC 3987.

— The option for media type to be an empty string has been removed, as this conflicts with the definition of media type in RFC 2046 and the existing regular expression defined in the schema referenced by C.2.

— Clause 7 (Physical package model, Clause 9 in the previous edition) has been slightly revised. Interleaving has been introduced before logical item names. Percent-encoding and un-percent encoding of non-ASCII characters have been explicitly introduced in 7.3.4 and 7.3.5.

— Clause 8 (Core properties, Clause 10 in the previous edition) has been rewritten by using prose and schemas rather than diagrams.

— Clause 10 (Digital signatures, Clause 12 in the previous edition) has been thoroughly revised. In particular, this clause now makes clear a convention for the choice of algorithms for signature and digest methods, which reflects the ongoing development of algorithms since the first edition of this document.

— Annex A has been made informative.

— The normative annex that defined pack IRIs (Annex B in the preceding editions) has been dropped. Pack IRIs are now defined in Clause 6.

— Annex C and Annex D (Annexes D and E in the preceding editions) no longer define schemas but reference externally defined schemas.

— Guidelines for meeting conformance requirements (Annex H in the preceding editions) have been dropped.

— Requirements around streaming consumption have been dropped.

— Wherever possible, requirements on programs have been rewritten as those on data.

— Annex H has been added to depict an example package.

— The Index (Annex J in the preceding editions) has been deleted.

— Bibliography has been added.

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.

Introduction

ISO/IEC 29500 (all Parts) 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.

This document includes two annexes (Annex C and Annex D) that refer to data files provided in electronic form.

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

This fourth edition preserves all previous functionality and adds no new functionality.

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.

Document description and processing languages — Office Open XML file formats —

Part 2:
Open packaging conventions

1 Scope

This document defines a set of conventions for packaging one or more interrelated byte streams (parts) as a single resource (package). These conventions are applicable not only to Office Open XML specifications as described in ISO/IEC 29500-1 and ISO/IEC 29500-4, but also to other markup specifications.

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.

ANSI/INCITS 4-1986 [R2017] - *Information Systems - Coded Character Sets - 7-Bit American National Standard Code For Information Interchange (7-Bit ASCII)*, American National Standards Institute (ANSI). 2017

FIPS 186-4, *Digital Signature Standard (DSS)*, National Institute of Standards and Technology, US Department of Commerce, July 2013

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

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

ISO 15836-1, *Information and documentation — The Dublin Core metadata element set — Part 1: Core elements*

ISO 15836-2, *Information and documentation — The Dublin Core metadata element set — Part 2: DCMI Properties and classes*

RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, The Internet Society, November 1996, N. Freed and N. Borenstein. Available at https://www.rfc-editor.org/info/rfc2046

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, The Internet Society, January 2005, Berners-Lee, T., R. Fielding, and L. Masinter. Available at https://www.rfc-editor.org/info/rfc3986

RFC 3987, *Internationalized Resource Identifiers (IRIs)*, The Internet Society, January 2005, Duerst, M. and M. Suignard. Available at https://www.rfc-editor.org/info/rfc3987

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*, The Internet Society, January 2008, D. Crocker and P.Overell, (editors). Available at https://www.rfc-editor.org/info/rfc5234

RFC 6931, *Additional XML Security Uniform Resource Identifiers (URIs)*, The Internet Society, April 2013, D. Eastlake 3rd. Available at https://www.rfc-editor.org/info/rfc6931

RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, The Internet Society, June 2014, R. Fielding and J. Reschke. Available at https://www.rfc-editor.org/info/rfc7231

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

The XML 1.0 specification, *Extensible Markup Language (XML) 1.0, Fourth Edition*. World Wide Web Consortium, 2006, Tim Bray, Jean Paoli, Eve Maler, C. M. Sperberg-McQueen, and François Yergeau (editors). Available at http://www.w3.org/TR/2006/REC-xml-20060816/[2)]

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

XML Base, *XML Base (Second Edition)*, World Wide Web Consortium, 28 January 2009. Jonathan Marsh and Richard Tobin (editors). Available at https://www.w3.org/TR/2009/REC-xmlbase-20090128/

W3C XML Schema Structures, *XML Schema Part 1: Structures (Second Edition)*, World Wide Web Consortium, 28 October 2004, Henry Thompson, David Beech, Murray Maloney and Noah Mendelsohn (editors). Available at https://www.w3.org/TR/xmlschema-1/

W3C XML Schema Datatypes, *XML Schema Part 2: Datatypes (Second Edition)*, World Wide Web Consortium, 28 October 2004, Paul Biron and Ashok Malhotra (editors). Available at https://www.w3.org/TR/xmlschema-2/

XML-Signature Syntax and Processing, World Wide Web Consortium, 12 February 2002, Donald Eastlake, Joseph Reagle and David Solo (editors). Available at http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

ZIP Appnote, *ZIP File Format Specification* Version 6.2.0, PKWARE Inc., 2004. Available at http://www.pkware.com/documents/APPNOTE/APPNOTE-6.2.0.txt

3 Terms and definitions

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

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1 Basics

3.1.1
byte
sequence of 8 bits treated as a unit

3.1.2
stream
linearly ordered sequence of *bytes* (3.1.1)

2) 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 document also makes normative reference, and which also depend upon XML, such as XML Namespaces and XML Base, are all aligned with the 5th Edition.

3.2 Abstract package model

3.2.1
part
stream (3.1.2) with a name, a MIME media type and associated common properties

3.2.2
abstract package
logical entity that holds a collection of *parts* (3.2.1) and *relationships* (3.2.3)

3.2.3
relationship
package relationship (3.2.4) or *part relationship* (3.2.5)

3.2.4
package relationship
connection from a package to a specific *part* (3.2.1) in the same package, or to an external resource

3.2.5
part relationship
connection from a *part* (3.2.1) in a package to another part in the same package, or to an external resource

3.2.6
source
part (3.2.1) or package from which a connection is established by a *relationship* (3.2.3)

3.2.7
target
part (3.2.1) or external resource to which a connection is established by a *relationship* (3.2.3)

3.2.8
relationship type
absolute IRI for specifying the role of a *relationship* (3.2.3)

3.2.9
Relationships part
part (3.2.1) containing an XML representation of *relationships* (3.2.3)

3.2.10
abstract package model
abstract model that defines *abstract packages* (3.2.2)

3.2.11
growth hint
suggested number of *bytes* (3.1.1) to reserve for a *part* (3.2.1) to grow in place

3.2.12
pack scheme
URI scheme that allows IRIs to be used as a uniform mechanism for addressing *parts* (3.2.1) within a package

3.2.13
pack IRI
IRI that conforms to the *pack scheme* (3.2.12)

3.2.14
part name
string that uniquely identifies a *part* (3.2.1) within a package

3.2.15
relationship identifier
string that consists of XML name characters and uniquely identifies a *relationship* (3.2.3) among those from the same *source* (3.2.6)

3.2.16
target mode
mode of resolution of relative references to *targets* (3.2.7)

3.2.17
I18N segment
Unicode string in a *part name* (3.2.14)

Note 1 to entry: The constraints on the value of the Unicode string shall be stated when the term is used in 6.2.2.2.

3.3 Physical package model

3.3.1
physical format
specific file format, or other persistence or transport mechanism

3.3.2
physical package
result of mapping an *abstract package* (3.2.2) to a *physical format* (3.3.1)

3.3.3
physical package model
pair of a *physical format* (3.3.1) and a mapping between the *abstract package model* (3.2.10) and that physical format

3.3.4
piece
portion of a *part* (3.2.1)

3.3.5
logical item
non-interleaved *part* (3.2.1), non-interleaved *Media Types stream* (3.3.12), *piece* (3.3.4) of an interleaved part, or piece of an interleaved Media Types stream

3.3.6
physical package item
atomic set of data in a *physical package* (3.3.2)

3.3.7
ZIP item
atomic set of data in a *ZIP file* (3.3.8) that becomes a file when the archive is uncompressed

3.3.8
ZIP file
file as defined in the ZIP Appnote

3.3.9
simple ordering
defined ordering for laying out the *parts* (3.2.1) in a package in which all the bits comprising each part are stored contiguously

3.3.10
interleaved ordering
defined ordering for laying out the *parts* (3.2.1) in a package in which parts are broken into *pieces* (3.3.4) and “mixed-in” with pieces from other parts

3.3.11
ASCII case-insensitive matching
comparing a sequence of code points as if all ASCII code points in the range 0x41 to 0x5A (A to Z) were mapped to the corresponding code points in the range 0x61 to 0x7A (a to z)

Note 1 to entry: The ASCII code points shall be as defined by ANSI/INCITS 4-1986.

3.3.12
Media Types stream
stream (3.1.2) in a *physical package* (3.3.2) representing an XML document that specifies the media type of each *part* (3.2.1) in the package

3.4 Digital signature and thumbnail

3.4.1
signature policy
specification of what *parts* (3.2.1) and *relationships* (3.2.3) are included in a signature and what additional behaviors are required for generating and validating signatures

3.4.2
thumbnail
small image that is a graphical representation of a *part* (3.2.1) or the package as a whole

3.5 Implementations

3.5.1
package implementer
software that implements physical input-output operations on a package according to the requirements and recommendations of this document

3.6 Core properties

3.6.1
core property
property of a package

4 Conformance

A package is of conformance class OPC if it obeys all syntactic constraints specified in this document.

OPC conformance is purely syntactic.

5 Overview

This document describes an abstract package model (Clause 6) and a physical package model (Clause 7) for the use of XML, Unicode, ISO/IEC 10646 (see Reference [7]), ZIP, and other relevant technologies and specifications to organize the content and resources of a document within a package. The package structure is intended to support the organization of constituent resources for various applications and categories of content. An example package is shown in Annex H.

The abstract package model is a package abstraction that holds a collection of parts and relationships. 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. Parts have MIME media types and are uniquely identified using the well-defined naming rules provided in this document.

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

This document also describes other features, including core properties for package metadata, a thumbnail for graphical representation of a package, and digital signatures of package contents. This document relies on ISO/IEC 29500-3 to allow future extensions of OPC without introducing compatibility problems.

This document specifies requirements for packages. Conformance requirements are identified throughout this document. A formal conformance statement is given in Clause 4.

6 Abstract package model

6.1 General

This clause introduces abstract packages in terms of parts (6.2) and relationships (6.5). It also introduces the pack scheme (6.3.2).

The purpose of an abstract package is to aggregate constituent components of a document (or other type of content) into a single object. For example, an abstract package holding a document with a picture can contain an XML markup part representing the text of the document and another part representing the picture.

An example abstract package is shown in H.2.

6.2 Parts

6.2.1 General

A part is analogous to a file in a file system or to a resource on an HTTP server.

6.2.2 Part names

6.2.2.1 General

A part shall have a part name, which shall uniquely identify a part within an abstract package.

6.2.2.2 Syntax

A part name shall be a Unicode string that matches the following production rules in the ABNF syntax defined in RFC 5234:

```
part_name = 1*( "/" isegment-nz )
isegment-nz = <isegment-nz, see RFC3987, Section 2.2>
```

and that further satisfies the constraints listed below, where an I18N segment is a Unicode string that matches the non-terminal `isegment-nz` and percent-encoding represents a character by the percent character "%" followed by two hexadecimal digits, as specified in RFC 3986:

— No I18N segments shall contain percent-encoded forward slash (“/”), or backward slash (“\”) characters.

— No I18N segments shall contain percent-encoded characters that match the non-terminal `iunreserved` in RFC 3987.

— No I18N segments shall end with a dot (“.”) character.

The part name "/_rels/.rels" shall be reserved (6.5.2.2). Part names in which the second-to-last I18N segment is equivalent to "_rels" and the final segment is equivalent to any string ending with ".rels" shall be reserved (6.5.2.3).

EXAMPLE 1 The part name "/hello/world/doc.xml" contains three path segments, namely, "hello", "world", and "doc.xml".

EXAMPLE 2 The part name "/é" contains a path segment "é" where é is 'LATIN SMALL LETTER E WITH ACUTE' (U+00E9).

NOTE Path segments are not explicitly represented as folders in the abstract package model; and no directory of folders exists in the abstract package model.

A package implementer is not required to support non-ASCII part names, although doing so is recommended.

6.2.2.3 Part name equivalence and integrity in an abstract package

Equivalence of part names shall be determined by ASCII case-insensitive matching. Such matching compares a sequence of code points as if all ASCII code points in the range 0x41–0x5A (A–Z) were mapped to the corresponding code points in the range 0x61–0x7A (a–z). See Reference [1].

The names of two different parts within an abstract package shall not be equivalent.

EXAMPLE 1 If an abstract package contains a part named "/a", the name of another part in that abstract package cannot be "/a" or "/A".

For each part name N and string S, let the result of concatenating N, the forward slash, and S be denoted by N[S]. A part name N1 is said to be derivable from another part name N2 if, for some string S, N1 is equivalent to N2[S].

EXAMPLE 2 "/a/b" is derivable from "/a", where N is "/a" and S is "b".

The name of a part shall not be derivable from the name of another part.

EXAMPLE 3 Suppose that an abstract package contains a part named "/segment1/segment2/…/segmentn". For it not to be derivable, other parts in that abstract package cannot have names such as "/segment1", "/SEGMENT1", "/segment1/segment2", "/segment1/SEGMENT2", or "/segment1/segment2/…/segmentn-*1*".

This subclause further introduces recommendations, so that Unicode Normalization Form C (NFC) and Unicode Normalization Form D (NFD) of part names do not cause part-name collisions. Note that some implementations of directory structures always apply NFD normalization.

The application of NFC or NFD normalization to the names of two different parts within an abstract package should not yield equivalent strings.

If an abstract package contains a part named "/é", where é is 'LATIN SMALL LETTER E' (U+0065) followed by 'COMBINING ACUTE ACCENT' (U+0301), the name of another part in that abstract package should not be "/é", where é is 'LATIN SMALL LETTER E WITH ACUTE' (U+00E9), or "/É", where É is 'LATIN CAPITAL LETTER E WITH ACUTE '(U+00C9).

If an abstract package contains a part named "/Å", where Å is 'ANGSTROM SIGN' (U+212B), the name of another part in that abstract package should not be "/Å" where Å is 'LATIN CAPITAL LETTER A WITH RING ABOVE' (U+00C5) because U+212B and U+00C5 are normalized to the same character sequence.

A part name N1 is said to be weakly derivable from another part name N2 if, for some string S, the result of applying NFC or NFD to N1 is equivalent to the result of applying NFC or NFD to N2[S].

EXAMPLE 4 Consider a part name "/é", where é is 'LATIN SMALL LETTER E WITH ACUTE' (U+00E9). Another part name "/é/a", where é is 'LATIN SMALL LETTER E' (U+0065) followed by 'COMBINING ACUTE ACCENT' (U+0301) is weakly derivable from "/é". Another part name "/É/a", where É is 'LATIN CAPITAL LETTER E' (U+0045) followed by 'COMBINING ACUTE ACCENT' (U+0301) is also weakly derivable.