TECHNICAL SPECIFICATION

ISO/IEC TS 30135-2

First edition
2014-11-15

Information technology — Digital publishing — EPUB3 —

Part 2: Publications

Technologies de l'information — Publications numériques — EPUB3 — Partie 2: Publications

Reference number
ISO/IEC TS 30135-2:2014(E)

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

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.

In other circumstances, particularly when there is an urgent market requirement for such documents, the joint technical committee may decide to publish an ISO/IEC Technical Specification (ISO/IEC TS), which represents an agreement between the members of the joint technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/IEC TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/IEC TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

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 TS 30135 series were prepared by Korean Agency for Technology and Standards (as KS X 6070 series) with International Digital Publishing Forum and were adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, Information technology, in parallel with its approval by the national bodies of ISO and IEC.

ISO/IEC TS 30135 consists of the following parts, under the general title *Information technology — Document description and processing languages — EPUB 3:*

— *Part 1: Overview*

— *Part 2: Publications*

— *Part 3: Content Documents*

— *Part 4: Open Container Format*

— *Part 5: Media Overlay*

— *Part 6: Canonical Fragment Identifier*

— *Part 7: Fixed-Layout Documents*

EPUB Publications 3.0

Recommended Specification 11 October 2011

THIS VERSION

http://www.idpf.org/epub/30/spec/epub30-publications-20111011.html

LATEST VERSION

http://www.idpf.org/epub/30/spec/epub30-publications.html

PREVIOUS VERSION

http://www.idpf.org/epub/30/spec/epub30-publications-20110908.html

A diff of changes from the previous draft is available at this link.

Please refer to the errata for this document, which may include some normative corrections.

Editors

Markus Gylling, DAISY Consortium

William McCoy, International Digital Publishing Forum (IDPF)

Matt Garrish, Invited Expert

TABLE OF CONTENTS

1 Overview

1.1 Purpose and Scope

This section is informative

This specification, EPUB Publications 3.0, defines publication-level semantics and conformance requirements for EPUB® 3, including the format of the Package Document and rules for how this document and other Publication Resources are associated to create a conforming EPUB Publication.

This specification is one of a family of related specifications that compose EPUB 3, the third major revision of an interchange and delivery format for digital publications based on XML and Web Standards. It is meant to be read and understood in concert with the other specifications that make up EPUB 3:

- The EPUB 3 Overview [EPUB3Overview], which provides an informative overview of EPUB and a roadmap to the rest of the EPUB 3 documents. The Overview should be read first.
- EPUB Content Documents 3.0 [ContentDocs30], which defines profiles of XHTML, SVG and CSS for use in the context of EPUB Publications.

- EPUB Open Container Format (OCF) 3.0 [OCF3], which defines a file format and processing model for encapsulating a set of related resources into a single-file (ZIP) EPUB Container.
- EPUB Media Overlays 3.0 [MediaOverlays30], which defines a format and a processing model for synchronization of text and audio.

This specification supersedes Open Package Format 2.0.1 [OPF2]. Refer to [EPUB3Changes] for information on differences between this specification and its predecessor.

1.2 Terminology

EPUB Publication (or Publication)

A logical document entity consisting of a set of interrelated resources and packaged in an EPUB Container, as defined by this specification and its sibling specifications.

Publication Resource

A resource that contains content or instructions that contribute to the logic and rendering of the EPUB Publication. In the absence of this resource, the Publication might not render as intended by the Author. Examples of Publication Resources include the Package Document, EPUB Content Documents, EPUB Style Sheets, audio, video, images, embedded fonts and scripts.

With the exception of the Package Document itself, Publication Resources must be listed in the manifest and must be bundled in the EPUB container file unless specified otherwise in Publication Resource Locations.

Examples of resources that are not Publication Resources include those identified by the Package Document link element and those identified in outbound hyperlinks that resolve outside the EPUB Container (e.g., referenced from an [HTML5] a element `href` attribute).

Foreign Resource

A Publication Resource that is not a Core Media Type. A Foreign Resource requires at least one fallback, as defined in Restrictions and Fallbacks.

Core Media Type Resource

A Publication Resource that is a Core Media Type and may therefore be included in the EPUB Publication without the provision of fallbacks.

EPUB Content Document

A Publication Resource that conforms to one of the EPUB Content Document definitions (XHTML or SVG).

An EPUB Content Document is a Core Media Type, and may therefore be included in the EPUB Publication without the provision of fallbacks.

XHTML Content Document

An EPUB Content Document conforming to the profile of [HTML5] defined in XHTML Content Documents [ContentDocs30].

XHTML Content Documents use the XHTML syntax of [HTML5].

SVG Content Document

An EPUB Content Document conforming to the constraints expressed in SVG Content Documents [ContentDocs30].

EPUB Navigation Document

A specialization of the XHTML Content Document, containing human- and machine-readable global navigation information, conforming to the constraints expressed in EPUB Navigation Documents [ContentDocs30].

Scripted Content Document

An EPUB Content Document that includes scripting or an XHTML Content Document that contains HTML5 forms elements.

Refer to Scripted Content Documents [ContentDocs30] for more information.

Top-level Content Document

An EPUB Content Document referenced directly from the spine

Core Media Type

A set of Publication Resource types for which no fallback is required. Refer to *Publication Resources* for more information.

Package Document

A Publication Resource carrying bibliographical and structural metadata about the EPUB Publication, as defined in *Package Documents*.

Manifestation

The digital (or physical) embodiment of a work of intellectual content. Changes to the content such as significant revision, abridgement, translation, or the realization of the content in a different digital or physical form result in a new manifestation. There may be many individual but identical copies of a manifestation, termed 'instances' or 'items'. The ISBN is an example of a manifestation identifier, and is shared by all instances of that manifestation.

All instances of a manifestation need not be bit-for-bit identical, as minor corrections or revisions are not judged to create a new manifestation or work.

Unique Identifier

The Unique Identifier is the primary identifier for an EPUB Publication, as identified by the `unique-identifier` attribute. The Unique Identifier may be shared by one or many Manifestations of the same work that conform to the EPUB standard and embody the same content, where the differences between the Manifestations are limited to those changes that take account of differences between EPUB Reading Systems (and which themselves may require changes in the ISBN).

The Unique Identifier is less granular than the ISBN. However, significant revision, abridgement, etc. of the content requires a new Unique Identifier.

Package Identifier

The Package Identifier allows any instance of an EPUB Publication to be compared against another to determine if they are identical, different versions of the same Manifestation, or unrelated.

Refer to Package Identifier for more information.

Manifest

A list of all Publication Resources that constitute the EPUB Publication.

Refer to manifest for more information.

Spine

An ordered list of Publication Resources, typically EPUB Content Documents, representing

the default reading order of the Publication.

Refer to spine for more information.

Media Overlay Document

An XML document that associates the XHTML Content Document with pre-recorded audio narration in order to provide a synchronized playback experience, as defined in [MediaOverlays30].

Text-to-Speech (TTS)

The rendering of the textual content of an EPUB Publication as artificial human speech using a synthesized voice.

EPUB Style Sheet (or Style Sheet)

A CSS Style Sheet conforming to the CSS profile defined in EPUB Style Sheets [ContentDocs30].

Viewport

The region of an EPUB Reading System in which the content of an EPUB Publication is rendered visually to a User.

CSS Viewport

A Viewport capable of displaying CSS-styled content.

EPUB Container (or Container)

The ZIP-based packaging and distribution format for EPUB Publications defined in [OCF3].

Author

The person(s) or organization responsible for the creation of an EPUB Publication, which is not necessarily the creator of the content and resources it contains.

User

An individual that consumes an EPUB Publication using an EPUB Reading System.

EPUB Reading System (or Reading System)

A system that processes EPUB Publications for presentation to a User in a manner conformant with this specification and its sibling specifications.

› 1.3 Conformance Statements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

All sections of this specification are normative except where identified by the informative status label "This section is informative". The application of informative status to sections and appendices applies to all child content and subsections they may contain.

All examples in this specification are informative.

› 2 EPUB Publications

This section defines conformance requirements for EPUB Publications and EPUB Reading Systems at the Publication level. Conformance requirements particular to specific Publication Resources and processing contexts are located in the specifications referenced herein.

› 2.1 Content Conformance

An EPUB Publication must meet all of the following criteria:

All Publication Resources

› All Publication Resources it contains must be represented in the Package Document (as defined in manifest), adhere to the constraints for Core Media Types and Fallback and be located as per Publication Resource Locations.

The Package Document

› It must contain exactly one Package Document, which must conform to the content requirements defined in Package Document — Content Conformance.

Content Documents

› It must contain at least one EPUB Content Document conformant to the content requirements defined in EPUB Content Documents [ContentDocs30].

The EPUB Navigation Document

› It must contain exactly one EPUB Navigation Document conformant to the content requirements defined in EPUB Navigation Documents — Content Conformance [ContentDocs30].

EPUB Style Sheets

› It may contain zero or more EPUB Style Sheets conformant to the content requirements defined in EPUB Style Sheets — Content Conformance [ContentDocs30].

EPUB Pronunciation Lexicons

› It may contain zero or more PLS Documents conformant to the content requirements defined in PLS Documents — Content Conformance [ContentDocs30].

Media Overlay Documents

› It may contain zero or more Media Overlay Documents conformant to the content requirements defined in [MediaOverlays30].

Additional Publication Resources

› It may contain zero or more Publication Resources in addition to those listed above, each of which must adhere to the requirements in All Publication Resources.

Container

› It must be packaged in a EPUB Container as defined in [OCF3].

› 2.2 Reading System Conformance

An EPUB Reading System must meet all of the following criteria:

EPUB 3 Processing

› It must process the EPUB Container as defined in [OCF3].

- It must process the Package Document as defined in Package Document — Reading System Conformance, and honor all presentation logic expressed through the Package Document (e.g., the reading order, fallback chains and bindings).

- It must not fail catastrophically if it encounters two distinct EPUB Publications with the same Unique Identifier.

- Unless specified as conditional behavior in this section, it must support all Core Media Type Resources.

- It may support an arbitrary set of Foreign Resource types, and must process fallbacks for unsupported Foreign Resources as defined in Restrictions and Fallbacks if not.

- It must process XHTML Content Documents as defined in XHTML Content Documents — Reading System Conformance [ContentDocs30].

- It must process SVG Content Documents as defined in SVG Content Documents — Reading System Conformance [ContentDocs30].

- If it has a CSS Viewport, it must support visual rendering of XHTML Content Documents as defined in EPUB Style Sheets — Reading System Conformance [ContentDocs30].

- If it has the capability to render raster images, it must support the raster image Core Media Types.

- If it has the capability to render vector images, it must support the vector image Core Media Types.

- If it has the capability to render pre-recorded audio, it must support the MP3 audio Core Media Type, should support the MP4 audio Core Media Type and should support Media Overlays [MediaOverlays30].

- If it supports Text-to-Speech (TTS) rendering, it should support PLS Documents [ContentDocs30], the CSS3 Speech features of the EPUB CSS Profile [ContentDocs30] and SSML attributes [ContentDocs30] in XHTML Content Documents.

- It must support the EPUB Canonical Fragment Identifiers scheme [EPUBCFI] for linking, and may support additional linking schemes as defined in the EPUB Linking Scheme Registry.

> NOTE
>
> It is recommended that Reading Systems support at least one of the [H.264] and [VP8] video codecs, but this is not a conformance requirement; a Reading System may support no video codecs at all. Content creators and Reading System developers should take into consideration factors such as breadth of adoption, video playback quality, and technology usage royalty requirements when making a choice to include or implement video in either (or potentially, both) formats.

Backward Compatibility

- It should process EPUB version 2 Publications as defined in [OPF2], [OPS2] and [OCF2].

- It must attempt to process any Publication whose Package Document `version` attribute designates a version lower than `3.0` or which omits the `version` attribute.

Forward Compatibility

- It should attempt to process any Publication whose Package Document `version` attribute designates a version higher than `3.0` or which omits the `version` attribute.

XML Processing

› It must be a conformant non-validating processor [XML].

› It must be a conformant processor as defined in [XMLNS].

› It must support `xml-stylesheet` processing instructions [ASSOCSS], and may support additional processing instructions.

› It must be a conformant application as defined by [XML Base].

> NOTE
>
> A conforming Reading System is not necessarily a single dedicated program or device, but may exist as a distributed system.

› 3 Package Documents

› 3.1 Introduction

This section is informative

The Package Document carries bibliographic and structural metadata about an EPUB Publication, and is thus the primary source of information about how to process and display it.

The Package Document is an XML document consisting of a set of container elements, each dedicated to housing information about a particular aspect of the Publication. These containers effectively centralize metadata for the Publication, detail the individual resources that compose it and provide reading order and other information for rendering the Publication to a User.

The following list summarizes the information a Package Document contains:

- Publication metadata — mechanisms for including and/or referencing metadata applicable to the entire Publication and particular resources within it.
- A Publication manifest — identifies (via IRI) and describes (via MIME media type) the set of resources that collectively compose the Publication.
- A spine — an ordered sequence of ID references to top-level resources in the manifest from which all other resources in the set can be reached or utilized. The spine defines the default reading order of the Publication.
- Fallback chains — an optional means for Publications to define an ordered list of top-level resources that can be considered content equivalents that a Reading System can choose between for rendering.
- Bindings — an optional means of associating script-based implementations with custom media types.

› 3.2 Content Conformance

A Package Document must meet all of the following criteria:

Document Properties

› It must meet the conformance constraints for XML documents defined in XML Conformance.

› It must be valid to the Package Document schema, as defined in Appendix A, *Package Document Schema*, and conform to all content conformance constraints expressed in Package Document Definition.

File Properties

› The Package Document filename should use the file extension `.opf`.

Package Documents have the MIME media type `application/oebps-package+xml` [RFC4839].

› 3.3 Reading System Conformance

An EPUB Reading System must meet all of the following criteria:

Processing

› It must process the Package Document in conformance with all Reading System conformance constraints expressed in Package Document Definition.

› 3.4 Package Document Definition

All elements [XML] defined in this section are in the `http://www.idpf.org/2007/opf` namespace [XMLNS] unless otherwise specified.

› 3.4.1 The `package` Element

The `package` element is the root container of the Package Document and encapsulates Publication metadata and resource information.

Element Name

`package`

Usage

The `package` element is the root element of the Package Document.

Attributes

`version` [required]

Specifies the EPUB specification version to which the Publication conforms.

The attribute must have the value `3.0` to indicate compliance with this version of the specification.

`unique-identifier` [required]

An IDREF [XML] that identifies the `dc:identifier` element that provides the package's preferred, or primary, identifier.

Refer to Publication Identifiers for more information.

`prefix` [optional]

Declaration mechanism for prefixes not reserved by this specification.

Refer to The `prefix` Attribute for more information.

`xml:lang` [optional]

Specifies the language used in the contents and attribute values of the carrying element and its descendants, as defined in section 2.12 Language Identification of [XML].

`dir` [optional]

Specifies the base text direction of the content and attribute values of the carrying element and its descendants.

Inherent directionality specified using [Unicode] takes precedence over this attribute.

Allowed values are `ltr` (left-to-right) or `rtl` (right-to-left).

`id` [optional]

The ID [XML] of this element, which must be unique within the document scope.

Content Model

In this order: `metadata` [required], `manifest` [required], `spine` [required], `guide` [optional/deprecated], `bindings` [optional]

› 3.4.2 The `metadata` Element

The `metadata` element encapsulates Publication meta information.

Element Name

`metadata`

Usage

Required first child of `package`.

Attributes

The `metadata` element has no attributes defined in this specification.

Content Model

In any order: `dc:identifier` [1 or more], `dc:title` [1 or more], `dc:language` [1 or more], `DCMES Optional Elements` [0 or more], `meta` [1 or more], `OPF2 meta` [0 or more], `link` [0 or more]

The minimal required metadata that Publications must include consists of three elements from the Dublin Core Metadata Element Set [DCMES] — `title`, `identifier` and `language` — together with the `modified` property from DCMI Metadata Terms [DCTERMS]. Refer to the example at the end of this section for an instance of a complete minimal metadata set.

Additional optional metadata is expressed using the DCMES optional elements and the `meta` element.

Examples

The following example represents the minimal set of metadata that all Publications must contain.

```
<package … unique-identifier="pub-id">
    …
```

```
    <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
        <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
A64CBE366809</dc:identifier>
        <dc:title>Norwegian Wood</dc:title>
        <dc:language>en</dc:language>
        <meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
    </metadata>
    …
</package>
```

› 3.4.3 The DCMES `identifier` Element

The [DCMES] `identifier` element contains a single identifier associated with the EPUB Publication, such as a UUID, DOI, ISBN or ISSN.

Element Name

`dc:identifier`

Namespace

`http://purl.org/dc/elements/1.1/`

Usage

Required child of `metadata`. Repeatable.

Attributes

`id` [optional]

The ID [XML] of this element, which must be unique within the document scope.

The `id` attribute is required on the `identifier` element containing the unique identifier. See below.

Content Model

Text

Every `metadata` section must include at least one `identifier` element containing an unambiguous identifier for the Publication. Multiple `identifier` elements are permitted, but only one can be marked as the Unique Identifier via the `package` element `unique-identifier` attribute.

The following example shows the unique `identifier` *element for a Publication.*

```
<package … unique-identifier="pub-id">
    <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
        <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
A64CBE366809</dc:identifier>
        …
    </metadata>
</package>
```

This specification makes a distinction between the Unique Identifier for an EPUB Publication and the identifier that uniquely identifies a specific version of it (i.e., to be able to differentiate EPUB Publications containing different versions of the same Manifestation). Two copies of an EPUB that are bit-for-bit

identical are the same version and must retain the same last modified date. If they are not bit-for-bit identical, they represent different versions, and must have different last modified dates.

To identify a specific version of a packaged Publication, a Package Identifier can be constructed by combining the Unique Identifier with the last modified date of the Publication. Changes between versions may include minor typographic or markup corrections, without affecting the Unique Identifier. Significant revisions to the content that result in a new edition require a change of the Unique Identifier. For more information on the semantics and requirements of the Package Identifier, refer to Package Identifier.

This specification imposes no additional restrictions or requirements on identifiers except that they must be at least one character in length. It is strongly recommended that all identifiers be fully qualified URIs, however.

Reading Systems must trim all leading and trailing whitespace from the element value, as defined by the XML specification [XML], before processing the value.

To determine whether an `identifier` conforms to an established system or has been granted by an issuing authority, Reading Systems should parse the value of the property. For additional precision (e.g., if the scheme cannot be determined from the value or could lead to an ambiguous result), Authors may attach an `identifier-type` property to assist in Reading System identification. When included, the `identifier-type` property should take precedence over value parsing the `identifier`.

The following example shows how an `identifier` *can be additionally marked as a DOI using the* `identifier-type` *property.*

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:identifier id="pub-
id">urn:doi:10.1016/j.iheduc.2008.03.001</dc:identifier>
    <meta refines="#pub-id" property="identifier-type"
scheme="onix:codelist5">06</meta>
    …
</metadata>
```


This specification does not require or endorse the use of any specific scheme for identifiers, and imposes no restrictions or requirements on `identifier-type` identifiers beyond those specified in the property definition.

When an EPUB Publication is derived from another publication, the identifier for that source publication may be included in the Publication metadata, and must be represented using the DCMES `source` element.

› 3.4.4 The DCMES `title` Element

The [DCMES] `title` element represents an instance of a name given to the EPUB Publication.

Element Name

`dc:title`

Namespace

`http://purl.org/dc/elements/1.1/`

Usage

Required child of `metadata`. Repeatable.

Attributes

`id` [optional]