INTERNATIONAL STANDARD

ISO/IEC 23736-2

First edition
2020-02

Information technology — Digital publishing — EPUB 3.0.1 —

Part 2: Publications

Technologies de l'information — Publications numériques — EPUB 3.0.1 —

Partie 2: Publications

Reference number
ISO/IEC 23736-2:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: 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.

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 (see www.iso.org/directives).

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 http://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.

This document was prepared by the World Wide Web Consortium (W3C) (as EPUB Publications 3.0.1) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 23736 series can be found on the ISO website.

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.

EPUB Publications 3.0.1

Recommended Specification 26 June 2014

THIS VERSION

http://www.idpf.org/epub/301/spec/epub-publications-20140626.html

LATEST VERSION

http://www.idpf.org/epub3/latest/publications

PREVIOUS VERSION

http://www.idpf.org/epub/301/spec/epub-publications-20140228.html

A diff of changes from the previous version is also available.

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

Editors

Markus Gylling, International Digital Publishing Forum (IDPF)

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 3.0.1, defines semantics and conformance requirements for EPUB® Publications, including the format of the Package Document that describes each Rendition of the content 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 3.0.1 [ContentDocs301], which defines profiles of XHTML, SVG and CSS for use in the context of EPUB Publications.
- EPUB Open Container Format (OCF) 3.0.1 [OCF301], 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 3.0.1 [MediaOverlays301], which defines a format and a processing model for synchronization of text and audio.

This specification supersedes EPUB Publications 3.0 [Publications30]. Refer to [EPUB3Changes] for information on differences between this specification and its predecessor.

› 1.2 Terminology

EPUB Publication

A collection of one or more Renditions conforming to this specification and its sibling specifications , packaged in an EPUB Container.

An EPUB Publication typically represents a single intellectual or artistic work, but this specification and its sibling specifications do not circumscribe the nature of the content.

Rendition

A logical document entity consisting of a set of interrelated resources representing one rendering of an EPUB Publication.

Default Rendition

The Rendition listed in the first `rootfile` element in the Container – META-INF/container.xml [OCF301] file.

Publication Resource

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

With the exception of the Package Document itself, the Publication Resources required to render a Rendition are listed in that Rendition's manifest and 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 [ContentDocs301] .

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 [ContentDocs301] .

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 [ContentDocs301] .

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 [ContentDocs301] for more information.

Top-level Content Document

An EPUB Content Document referenced from the spine, whether directly or via a fallback chain [Publications301] .

Fixed-Layout Document

An EPUB Content Document directly referenced from the spine that has been designated `pre-paginated` in the Package Document, as defined in The rendition:layout Property [Publications301] .

The dimensions to use for rendering Fixed-Layout Documents are defined in Fixed-Layout Documents [ContentDocs301] .

Synthetic Spread

The rendering of two adjacent pages simultaneously on a device screen.

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 a given Rendition of an EPUB Publication, as defined in *Package Documents* .

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 Renditions of the same EPUB Publication that conform to the EPUB standard and embody the same content.

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

Release Identifier

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

Refer to Release Identifier for more information.

Manifest

A list of all Publication Resources that constitute the given Rendition of a 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 given Rendition of an EPUB 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 [MediaOverlays301].

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 [ContentDocs301] .

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 [OCF301].

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 .

User Agent

A client or application that consumes generic HTML (e.g., Web browser, screen readers)

› 1.3 Typographic Conventions

The following typographic conventions are used in this specification:

`markup`

All markup (elements, attributes, properties), code (JavaScript, pseudo-code), machine processable values (string, characters, media types) and file names are in red-orange monospace font.

markup

Links to markup and code definitions are underlined and in red-orange monospace font. Only the first instance in each section is linked.

http://www.idpf.org/

URIs are in navy blue monospace font.

hyperlink

Hyperlinks are underlined and in blue.

[reference]

Normative and informative references are enclosed in square brackets.

Term

Terms defined in the Terminology are in capital case.

Term

Links to term definitions have a dotted blue underline. Only the first instance in each section is linked.

Normative element, attribute and property definitions are in blue boxes.

```
Informative markup examples are in white boxes.
```

NOTE

Informative notes are in yellow boxes with a "Note" header.

CAUTION

Informative cautionary note are in red boxes with a "Caution" header.

› 1.4 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 Rendition level. Conformance requirements particular to specific Publication Resources and processing contexts are located in the specifications referenced herein.

› 2.1 Content Conformance

Each Rendition of an EPUB Publication MUST meet all of the following criteria:

All Publication Resources

› All Publication Resources MUST be listed 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 [ContentDocs301] .

The EPUB Navigation Document

› It MUST contain exactly one EPUB Navigation Document conformant to the content requirements defined in EPUB Navigation Documents — Content Conformance [ContentDocs301] .

EPUB Style Sheets

› It MAY contain zero or more EPUB Style Sheets conformant to the content requirements defined in EPUB Style Sheets — Content Conformance [ContentDocs301] .

EPUB Pronunciation Lexicons

› It MAY contain zero or more PLS Documents conformant to the content requirements defined in PLS Documents — Content Conformance [ContentDocs301] .

Media Overlay Documents

› It MAY contain zero or more Media Overlay Documents conformant to the content requirements defined in [MediaOverlays301].

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 [OCF301].

› 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 [OCF301].
- 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, bindings, page progression direction and fixed layouts).
- 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 [ContentDocs301] .
- It MUST process SVG Content Documents as defined in SVG Content Documents — Reading System Conformance [ContentDocs301] .
- If it has a CSS Viewport , it MUST support visual rendering of XHTML Content Documents as defined in EPUB Style Sheets — Reading System Conformance [ContentDocs301] .
- 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 [MediaOverlays301].
- If it supports Text-to-Speech (TTS) rendering, it SHOULD support PLS Documents [ContentDocs301] , the CSS3 Speech features of the EPUB CSS Profile [ContentDocs301] and SSML attributes [ContentDocs301] 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 given Rendition of an EPUB 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 given Rendition of an EPUB Publication whose Package Document `version` attribute designates a version higher than "`3.0`".

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 a Rendition of an EPUB Publication, and is thus the primary source of information about how to process and display that Rendition.

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 Rendition. These containers effectively centralize metadata, detail the individual resources that compose the Rendition and provide reading order and other information for rendering the EPUB Publication is represents to a User.

The following list summarizes the information a Package Document contains:

- Rendition metadata — mechanisms for including and/or referencing metadata applicable to the EPUB Publication and/or the specific Rendition of it, including for particular resources within the Rendition.
- A manifest — identifies (via IRI) and describes (via MIME media type) the set of resources that collectively compose the given Rendition of the EPUB 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 given Rendition.
- Fallback chains — an optional means for defining 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.

› It SHOULD process presentational metadata, as expressed in General Properties

› It MUST process fixed layout metadata, as expressed in Fixed-Layout Properties

- For fixed layouts expressed using the `rendition:layout` property, it MUST determine the rendering dimensions as defined in Fixed-Layout Documents [ContentDocs301] .

- It MUST ignore proprietary metadata properties that pertain to layout expressions if they conflict behaviorally with the property semantics defined in Fixed-Layout Properties.

› 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 metadata and resource information for a Rendition.

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 given Rendition 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) and `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], `collection` [0 or more]

› 3.4.2 The `metadata` Element

The `metadata` element encapsulates meta information for the given Rendition.

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 each Rendition of an EPUB Publication 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 Renditions have to 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 given Rendition of 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 Rendition's metadata section MUST include at least one identifier element containing an unambiguous identifier for the EPUB 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 an EPUB 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 different versions of the same EPUB Publication). Whenever a Rendition is modified, it MUST include a new last modified date.

To identify a specific version of a packaged EPUB Publication, a Release Identifier can be constructed by combining the Unique Identifier with the last modified date of the Rendition. For more information on the semantics and requirements of the Release Identifier, refer to Release Identifier.

Identifiers MAY differ from Rendition to Rendition.

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 EPUB 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]

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

`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) and `rtl` (right-to-left).

Content Model

Text

Every `metadata` section MUST include at least one `title` element containing the title for the EPUB Publication. Multiple `title` elements are permitted, but the `title-type` property SHOULD be attached to indicate the type of title (e.g., the main title of a work or a subtitle).

The following example shows how to indicate different title types.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:title id="t1">A Dictionary of Modern English Usage</dc:title>
    <meta refines="#t1" property="title-type">main</meta>

    <dc:title id="t2">First Edition</dc:title>
    <meta refines="#t2" property="title-type">edition</meta>

    <dc:title id="t3">Fowler's</dc:title>
    <meta refines="#t3" property="title-type">short</meta>
    …
</metadata>
```

When adding the `title-type` property, Authors SHOULD designate only one `title` element as containing the main title for the Publication. If no means of determining title types is provided, or understood, Reading Systems MUST treat the first `title` element in document order as the main title. This specification does not define how additional `title` elements SHOULD be processed in such situations.

The optional `display-seq` property MAY also be attached to each `title` to indicate their primacy for display and other rendering purposes.

The following example shows how to indicate display sequence.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:title id="t1">The Red and the Black</dc:title>
    <meta refines="#t1" property="title-type">main</meta>
    <meta refines="#t1" property="display-seq">1</meta>

    <dc:title id="t2">A Chronicle of the Nineteenth Century</dc:title>
    <meta refines="#t2" property="title-type">subtitle</meta>
    <meta refines="#t2" property="display-seq">2</meta>

    <dc:title id="t3">A Chronicle of 1830</dc:title>
    <meta refines="#t3" property="title-type">subtitle</meta>
    <meta refines="#t3" property="display-seq">3</meta>
    …
</metadata>
```

Titles MAY differ from Rendition to Rendition.

This specification imposes no additional restrictions or requirements on titles except that they MUST be at least one character in length.

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

› Examples

The following example shows how the title "THE LORD OF THE RINGS, Part One: The Fellowship of the Ring" could be classified.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title id="t1">The Fellowship of the Ring</dc:title>
    <meta refines="#t1" property="title-type">main</meta>

    <dc:title id="t2">The Lord of the Rings</dc:title>
    <meta refines="#t2" property="title-type">collection</meta>

    <dc:title id="t3">THE LORD OF THE RINGS, Part One: The Fellowship of
the Ring</dc:title>
    <meta refines="#t3" property="title-type">expanded</meta>
    …
</metadata>
```

The following example shows how the complex title "The Great Cookbooks of the World: Mon premier guide de cuisson, un Mémoire. The New French Cuisine Masters, Volume Two. Special Anniversary Edition" could be classified.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title id="t1" xml:lang="fr">Mon premier guide de cuisson, un
Mémoire</dc:title>
    <meta refines="#t1" property="title-type">main</meta>
    <meta refines="#t1" property="display-seq">2</meta>

    <dc:title id="t2">The Great Cookbooks of the World</dc:title>
    <meta refines="#t2" property="title-type">collection</meta>
    <meta refines="#t2" property="display-seq">1</meta>

    <dc:title id="t3">The New French Cuisine Masters</dc:title>
    <meta refines="#t3" property="title-type">collection</meta>
    <meta refines="#t3" property="display-seq">3</meta>

    <dc:title id="t4">Special Anniversary Edition</dc:title>
    <meta refines="#t4" property="title-type">edition</meta>
    <meta refines="#t4" property="display-seq">4</meta>

    <dc:title id="t5">The Great Cookbooks of the World:
        Mon premier guide de cuisson, un Mémoire.
        The New French Cuisine Masters, Volume Two.
        Special Anniversary Edition</dc:title>
    <meta refines="#t5" property="title-type">expanded</meta>
    …
</metadata>
```

› 3.4.5 The DCMES `language` Element

The [DCMES] `language` element specifies the language of the content of the given Rendition.

Element Name

`dc:language`

Namespace

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

Usage

Required child of `metadata` .

Attributes

`id` [optional]

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

Content Model

Text

Each Rendition's `metadata` section MUST include at least one `language` element with a value conforming to [RFC5646].

The following example shows an EPUB Publication is in U.S. English.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:language>en-US</dc:language>
    …
</metadata>
```

Additional language elements MAY be included for multilingual Publications, but each element's value MUST conform to [RFC5646].

Languages MAY differ from Rendition to Rendition.

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

› 3.4.6 The DCMES Optional Elements

All elements from the [DCMES] element set — except for identifier , language and title , as defined above — are designated as optional. These elements all conform to the following generalized definition:

Element Name

contributor | coverage | creator | date | description | format | publisher | relation | rights | source | subject | type

Namespace

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

Usage

Optional child of metadata . Repeatable.

Attributes

id [optional]

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

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) and rtl (right-to-left).

Content Model

Text

* The `xml:lang` and `dir` attributes are permitted only on the following elements: `contributor`, `coverage`, `creator`, `description`, `publisher`, `relation`, `rights` and `subject`.

Optional metadata MAY differ from Rendition to Rendition.

The value of all optional [DCMES] elements MUST be at least one character in length.

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

Except as detailed below, this specification does not modify the [DCMES] definitions for these elements.

› The DCMES `contributor` Element

The `contributor` element is used to represent the name of a person, organization, etc. that played a secondary role in the creation of the content of an EPUB Publication.

The use of the `contributor` element is identical to the use of the `creator` element in all other respects, as detailed in the next section.

› The DCMES `creator` Element

The `creator` element represents the name of a person, organization, etc. responsible for the creation of the content of an EPUB Publication. The `role` property can be attached to the element to indicate the function the creator played in the creation of the content.

The following example shows how to represent a `creator` as an author using a MARC relators term.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:creator id="creator">Haruki Murakami</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators"
id="role">aut</meta>
    …
</metadata>
```

The `creator` element SHOULD contain the name of the creator as a Reading System will present it to a User. The `file-as` property MAY be attached to include a normalized form of the name, and the `alternate-script` property can be used to represent a creator's name in another language or script.

The following example shows the different ways a creator's name can be included to facilitate processing and rendering.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:creator id="creator">Haruki Murakami</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators"
id="role">aut</meta>
```

```
    <meta refines="#creator" property="alternate-script" xml:lang="ja">村
上 春樹</meta>
    <meta refines="#creator" property="file-as">Murakami, Haruki</meta>
    …
</metadata>
```

If an EPUB Publication has more than one creator, each SHOULD be included in a separate `creator` element. The order in which to render the `creator` names SHOULD be specified using the `display-seq` property.

The following example shows how to indicate the display order for `creator` elements.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:creator id="creator01">Lewis Carroll</dc:creator>
    <meta refines="#creator01" property="role"
scheme="marc:relators">aut</meta>
    <meta refines="#creator01" property="display-seq">1</meta>

    <dc:creator id="creator02">John Tenniel</dc:creator>
    <meta refines="#creator02" property="role"
scheme="marc:relators">ill</meta>
    <meta refines="#creator02" property="display-seq">2</meta>
    …
</metadata>
```

If no means of establishing the primacy of creators for rendering is identifiable, Reading Systems MUST use the document order of `creator` elements in the `metadata` section (i.e., where the first `creator` element encountered is the primary creator). If a Reading System exposes creator metadata for the User, it SHOULD include all the creators listed in the `metadata` section whenever possible (e.g., when not constrained by display considerations).

Secondary contributors SHOULD be represented using DCMES `contributor` elements.

› The DCMES `date` Element

The `date` element MUST only be used to define the publication date of the EPUB Publication. The publication date is not the same as the last modified date (the last time the Rendition was changed), which MUST be included using the [DCTERMS] `modified` property.

It is RECOMMENDED that the date string conform to [ISO8601], particularly the subset expressed in W3C Date and Time Formats [DateTime], as such strings are both human and machine readable.

The following example shows a publication date.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:date>2000-01-01T00:00:00Z</dc:date>
    …
</metadata>
```

Additional dates SHOULD be expressed using the specialized date properties available in the [DCTERMS] vocabulary, or similar.

The publication date MAY be common to all instances of an EPUB Publication or MAY change from instance to instance (e.g., if the EPUB Publication gets generated on demand).

Only one `date` element is allowed.

› The DCMES `source` Element

The `source` element identifies the related resource(s) from which this EPUB Publication is derived.

When a Rendition includes page identifiers, using the `pagebreak` property from the EPUB Structural Semantics Vocabulary [StructureVocab], a `source-of` refinement property SHOULD be attached to the `source` element that identifies the source of the pagination.

The following example shows the ISBN identifier for an EPUB Publication together with the source ISBN identifier for the print work it was derived from.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:identifier id="isbn-id">urn:isbn:9780101010101</dc:identifier>
    <meta refines="#isbn-id" property="identifier-type"
scheme="onix:codelist5">15</meta>

    <dc:source id="src-id">urn:isbn:9780375704024</dc:source>
    <meta refines="#src-id" property="identifier-type"
scheme="onix:codelist5">15</meta>
    <meta refines="#src-id" property="source-of">pagination</meta>
    …
</metadata>
```

› The DCMES `type` Element

The `type` element is used to indicate that the given EPUB Publication is of a specialized type (e.g., annotations packaged in EPUB format or a dictionary).

The IDPF maintains an informative registry of specialized EPUB Publication types for use with this element at http://www.idpf.org/epub/vocab/package/types .

› 3.4.7 The `meta` Element

The `meta` element provides a generic means of including package metadata, allowing the expression of primary metadata about the package or content and refinement of that metadata.

Element Name

`meta`

Usage

As child of the `metadata` element. Repeatable.

Attributes

`property` [required]

A property.

Refer to Vocabulary Association Mechanisms for more information.

`refines` [context dependent]

Identifies the expression or resource augmented by this element. The value of the attribute MUST be a relative IRI [RFC3987] referencing the resource or element it describes.

The `refines` attribute is optional depending on the type of metadata being expressed. When omitted, the `meta` element defines a primary expression.

`id` [optional]

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

`scheme` [optional]

A property data type value indicating the source the value of the element is drawn from.

`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) and `rtl` (right-to-left).

Content Model

Text

Each `meta` element defines a metadata expression, where the `property` attribute defines the statement being made in the expression and the text content of the element represents the assertion.

This specification defines two types of metadata expressions that can be defined using the `meta` element:

- A *primary expression* is one in which the expression defined in the `meta` element establishes some aspect of the EPUB Publication. A `meta` element that omits a `refines` attribute defines a primary expression.
- A *subexpression* is one in which the expression defined in the `meta` element enhances the meaning of the expression or resource referenced in its `refines` attribute. A subexpression MAY

refine a media clip, for example, by expressing its duration, or refine a creator or contributor expression by defining the person's role.

Subexpressions are not limited to refining only primary expressions and resources; they MAY be used to refine the meaning of other subexpressions, thereby creating chains of information.

NOTE

All of the [DCMES] elements represent primary expressions, and permit refinement by `meta` element subexpressions.

This specification reserves a set of vocabularies for use in the `property` attribute, but terms from any vocabulary MAY be used so long as a prefix is declared for the vocabulary.

The `scheme` attribute can be used to identify the system or scheme that a `meta` element's value is drawn from. The value of the `scheme` attribute is a property data type that resolves to the resource that defines the scheme.

The following example shows how a subexpression can be attached to an `creator` to indicate it represents an author. The `scheme` indicates the value is drawn from the MARC relators terms.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:creator id="creator">Haruki Murakami</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators"
id="role">aut</meta>
    …
</metadata>
```

If a Reading System does not recognize the `scheme` attribute value, it SHOULD treat the value of the element as a string.

Reading Systems SHOULD ignore all `meta` elements whose `property` attributes define expressions they do not recognize. A Reading System MUST NOT fail when encountering unknown expressions.

In order to ensure that a Release Identifier can be constructed, the `metadata` element MUST contain exactly one `meta` element defining a [DCTERMS] `modified` property for the given Rendition. Additional `modified` properties MAY be included, but they MUST have a different subject (i.e., they MUST include a `refines` attribute that references an element or resource).

Every `meta` element MUST express a value that is at least one character in length after whitespace normalization.

Unless an individual property explicitly defines a different whitespace normalization algorithm, Reading Systems MUST trim all leading and trailing whitespace from the `meta` element values, as defined by the XML specification [XML], before further processing them.

› Examples

The following example represents a more complete set of metadata that a Rendition will typically contain.

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    …
    <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
A64CBE366809</dc:identifier>
    <meta refines="#pub-id" property="identifier-type"
scheme="xsd:string">uuid</meta>

    <dc:identifier id="isbn-id">urn:isbn:9780101010101</dc:identifier>
    <meta refines="#isbn-id" property="identifier-type"
scheme="onix:codelist5">15</meta>

    <dc:source id="src-id">urn:isbn:9780375704024</dc:source>
    <meta refines="#src-id" property="identifier-type"
scheme="onix:codelist5">15</meta>

    <dc:title id="title">Norwegian Wood</dc:title>
    <meta refines="#title" property="title-type">main</meta>

    <dc:language>en</dc:language>

    <dc:creator id="creator">Haruki Murakami</dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators"
id="role">aut</meta>
    <meta refines="#creator" property="alternate-script" xml:lang="ja">村
上 春樹</meta>
    <meta refines="#creator" property="file-as">Murakami, Haruki</meta>

    <meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>

</metadata>
```

The following example shows an identifier that has been issued by a metadata authority.

```
<package version="3.0"
         unique-identifier="pub-id"
         xmlns="http://www.idpf.org/2007/opf">
    <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
        <dc:identifier id="pub-id">urn:uuid:1234-5678</dc:identifier>
        <dc:identifier id="isbn-
id">urn:isbn:9780101010101</dc:identifier>

        <meta refines="#isbn-id" property="meta-auth" id="meta-
authority-01">Metadata Authority Inc.</meta>
        <link refines="#meta-authority-01" rel="xml-signature"
href="../META-INF/Signatures.xml#MAI-Signature"/>
        …
    </metadata>
</package>

<!-- in Signatures.xml -->
<signatures>
    <Signature Id="MAI-Signature"
xmlns="http://www.w3.org/2000/09/xmldsig#">
        …
    </Signature>
</signatures>
```

› 3.4.8 The meta Element (OPF2) [OBSOLETE]

The meta element defined in [OPF2] has been obsoleted and replaced by the new meta element, but MAY be included as an optional repeatable child of the metadata element for forwards compatibility purposes.

EPUB 3 Reading Systems MUST ignore this element.

› 3.4.9 The link Element

The link element is used to associate resources with the given Rendition of the EPUB Publication, such as metadata records.

Element Name

link

Usage

As a child of metadata . Repeatable.

Attributes

href [required]

An absolute or relative IRI reference [RFC3987] to a resource.

rel [required]

A space-separated list of property values.

id [optional]

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

refines [optional]

Identifies the expression or resource augmented by this element. The value of the attribute MUST be a relative IRI [RFC3987] referencing the resource or element it describes.

When the refines attribute is omitted, the expression applies to the EPUB Publication as a whole.

media-type [optional]

A media type [RFC2046] that specifies the type and format of the resource referenced by this link.

Content Model

Empty

The metadata element MAY contain zero or more link elements.

The href attribute of the link element identifies the location of the resource — inclusion of which is optional in the container file — and the rel attribute defines the nature of the resource (i.e., its relation to the given Rensition of the EPUB Publication or property specified in the refines attribute). Reading Systems are NOT REQUIRED to dereference these resources. Refer to Metadata link Properties for the list of resource types that are recognized by this specification.

Resources identified by the link element href attribute MUST NOT be listed in the manifest.

When the link element references a metadata record, precedence MUST be given to metadata defined inline in the Package Document metadata element in the case of conflicts.

The optional refines attribute can be attached when the referenced resource applies to another metadata item (e.g., to tie an XML Signature [XML DSIG Core] to a metadata authority). The resource applies to the Rendition when the attribute is not present.

If a Reading System does not recognize the relationship of the resource as defined in the rel attribute, it SHOULD ignore the link element.

› Examples

The following example shows the link element used to associate three metadata resources with the Rendition: an ONIX record, an XMP record, and a link to an informational web page. Note that as foaf is not a predefined prefix, the metadata extensibility mechanism is employed to associate the vocabulary.

```
<package … prefix="foaf: http://xmlns.com/foaf/spec/">
    <metadata>
        …
        <link rel="onix-record"
href="http://example.org/onix/12389347"/>
        <link rel="xmp-record" href="http://example.org/xmp/12389347"/>
        <link rel="foaf:homepage" href="http://example.org/book-
info/12389347" />
        …
    </metadata>
    …
</package>
```

› 3.4.10 The manifest Element

The manifest element provides an exhaustive list of the Publication Resources that constitute the given Rendition, each represented by an item element.

Element name

manifest

Usage

Required second child of package , following metadata .

Attributes

id [optional]

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

Content Model

One or more item elements [required]

NOTE

This specification supports internationalized resource naming, so elements and attributes that reference Publication Resources accept IRIs as their value. For compatibility with older Reading Systems that only accept URIs, resource names should be restricted to the ASCII character set.

› 3.4.11 The item Element

The item element represents a Publication Resource.

Element Name

item

Usage

As a child of manifest . Repeatable.

Attributes

id [required]

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

href [required]

An IRI [RFC3987] specifying the location of the Publication Resource described by this item.

media-type [required]

A media type [RFC2046] that specifies the type and format of the Publication Resource described by this item.

fallback [conditionally required]

An IDREF [XML] that identifies the fallback for a non-Core Media Type.

Refer to Manifest Fallbacks for more information.

properties [optional]

A space-separated list of property values.

Refer to Manifest item Properties for a set of properties defined by this specification.

media-overlay [optional]

An IDREF [XML] that identifies the Media Overlay Document for the resource described by this item.

Refer to Packaging [MediaOverlays301] for more information.

Content Model

Empty

Each item element in the manifest identifies a Publication Resource by the IRI provided in its href attribute. The IRI MAY be absolute or relative. In the case of relative IRIs, Reading Systems MUST use the IRI of the Package Document as the base when resolving these to absolute IRIs. The resulting absolute IRI MUST be unique within the manifest scope.

All Publication Resources MUST be referenced from the manifest, regardless of whether they are included in the EPUB Container or made available remotely. Refer to Publication Resource Locations for media type-specific requirements regarding resource locations.

The Publication Resource identified by an item element MUST conform to the applicable specification(s) as inferred from the MIME media type provided in the media-type attribute. Core Media Type Resources MUST use the media type designated in EPUB Core Media Types.

All Foreign Resources MUST provide a fallback as defined in Restrictions and Fallbacks.

All Publication Resources MUST declare any applicable descriptive metadata properties as defined in Manifest item Properties via the item element properties attribute. Exactly one item MUST be declared as the EPUB Navigation Document using the nav property.

Reading Systems MUST ignore all descriptive metadata properties that they do not recognize.

The manifest is not self-referencing: it MUST NOT include an item element that refers to the Package Document itself.

NOTE

The order of item elements in the manifest is not significant. The presentation sequence of content documents is provided in the spine .

› Examples

The following example shows a manifest that only contains Core Media Type Resources.

```
<manifest>
    <item id="nav"
          href="nav.xhtml"
          properties="nav"
          media-type="application/xhtml+xml"/>
    <item id="intro"
          href="intro.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c1"
          href="chap1.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c1-answerkey"
          href="chap1-answerkey.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c2"
          href="chap2.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c2-answerkey"
          href="chap2-answerkey.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c3"
          href="chap3.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="c3-answerkey"
          href="chap3-answerkey.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="notes"
          href="notes.xhtml"
          media-type="application/xhtml+xml"/>
    <item id="cover"
          href="./images/cover.svg"
          properties="cover-image"
          media-type="image/svg+xml"/>
    <item id="f1"
          href="./images/fig1.jpg"
          media-type="image/jpeg"/>
    <item id="f2"
          href="./images/fig2.jpg"
          media-type="image/jpeg"/>
    <item id="css"
          href="./style/book.css"
          media-type="text/css"/>
    <item id="pls"
          href="./speech/dict.pls"
          media-type="application/pls+xml"/>
</manifest>
```

The following example shows a **manifest** *that references two Foreign Resources, and therefore uses the fallback chain mechanism to supply content alternatives. The fallback chain terminates with a Core Media Type.*

```
<manifest>
    <item id="item1"
          href="chap1_docbook.xml"
          media-type="application/docbook+xml"
          fallback="fall1"/>
    <item id="fall1"
          href="chap1.xml"
          media-type="application/z3986-auth+xml"
          fallback="fall2" />
    <item id="fall2"
          href="chap1.xhtml"
          media-type="application/xhtml+xml"/>
    …
</manifest>
```

NOTE

Refer also to the Manifest item properties examples for use of the properties attribute.

› 3.4.12 The spine Element

The spine element defines the default reading order of the given Rendition of the EPUB Publication content by defining an ordered list of manifest item references.

Element name

spine

Usage

Required third child of package , following manifest .

Attributes

id [optional]

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

toc [optional]

An IDREF [XML] that identifies the manifest item that represents the superseded NCX.

Refer to NCX Superseded for more information.

page-progression-direction [optional]

The global direction in which the content flows.

Allowed values are ltr (left-to-right), rtl (right-to-left) and default.

When the default value is specified, the Author is expressing no preference and the Reading System MAY chose the rendering direction. This value MUST be assumed when the attribute is not specified.

Content Model

One or more itemref elements [required]

The spine represents an ordered subset of the Publication Resources listed in the manifest, providing the default reading order for the given Rendition.

Reading Systems MUST provide a means of rendering the Rendition in the order defined in the spine, which includes: 1) recognizing the first primary itemref as the beginning of the default reading order; and, 2) rendering successive primary items in the order given in the spine.

All EPUB Content Documents that are linked to from EPUB Content Documents in the spine MUST themselves be listed in the spine. Linked documents include documents referenced from the href attribute of a and area elements and scripted links (e.g., using DOM Events and/or form elements). All EPUB Content Documents linked to from the EPUB Navigation Document MUST be listed in the spine, as well, regardless of whether the Navigation Document has been included in the spine. The requirement to list linked documents applies recursively (i.e., all Content Documents linked to from linked Content Documents also have to be listed, and so on.).

Although the page-progression-direction attribute sets the global flow direction, individual Content Documents and parts of Content Documents MAY override this setting (e.g., via the writing-mode CSS property). Reading Systems MAY also provide mechanisms to override the default direction (e.g., buttons or settings that allow the application of alternate style sheets).

Reading Systems MUST ignore the page progression direction defined in pre-paginated XHTML Content Documents. The page-progression-direction attribute defines the flow direction from one fixed-layout page to the next.

› NCX Superseded

The NCX feature defined in [OPF2] is superseded by the EPUB Navigation Document [ContentDocs301] . EPUB 3 Publications MAY include an NCX (as defined in OPF 2.0.1) for EPUB 2 Reading System forwards compatibility purposes, but EPUB 3 Reading Systems MUST ignore the NCX.

> *NOTE*
>
> As the EPUB 2 NCX and the EPUB 3 Navigation Document use different mechanisms for identification in the Package Document (the spine toc attribute and the nav property on the manifest item element, respectively) they can co-exist without conflict in an EPUB 3 Publication.

› 3.4.13 The itemref Element

The child itemref elements of the spine represent a sequential list of Publication Resources (typically EPUB Content Documents). The order of the itemref elements defines the default reading order of the given Rendition of the EPUB Publication.

Element Name

itemref

Usage

As a child of spine . Repeatable.

Attributes

idref [required]

An IDREF [XML] that identifies a manifest item.

linear [optional]

Specifies whether the referenced content is primary.

The value of the attribute MUST be yes or no. The default value is yes.

id [optional]

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

properties [optional]

A space-separated list of property values.

Refer to Spine itemref Properties for a set of properties defined by this specification.

Content Model

Empty

Each itemref element MUST reference a unique item in the manifest via its idref attribute.

Each referenced manifest item MUST be either a) an EPUB Content Document or b) another type of Publication Resource which, *regardless of whether it is a Core Media Type Resource or a Foreign Resource* , MUST include an EPUB Content Document in its fallback chain.

> *NOTE*
>
> Although the EPUB Navigation Document is required in EPUB Publications, it is optional to include it in the spine.

The itemref element linear attribute indicates whether referenced item is considered primary (yes) or auxiliary (no) in the spine. Each Rendition MUST include at least one primary itemref.

The linear attribute MAY be used to enable Reading Systems to distinguish presentation of body content from supplementary content which might be, for example, presented in a popup window or omitted from an aural rendering. Reading Systems SHOULD provide Users the ability to control whether non-linear content is rendered in the default reading order.

Any applicable descriptive metadata properties, such as those defined in the Spine itemref Properties, SHOULD be declared via the properties attribute.

Reading Systems MUST ignore all metadata properties expressed in the properties attribute that they do not recognize.

› Examples

The following example shows a spine element corresponding to the manifest example above.

```
<spine page-progression-direction="ltr">
    <itemref idref="intro"/>
    <itemref idref="c1"/>
    <itemref idref="c1-answerkey" linear="no"/>
    <itemref idref="c2"/>
    <itemref idref="c2-answerkey" linear="no"/>
    <itemref idref="c3"/>
    <itemref idref="c3-answerkey" linear="no"/>
    <itemref idref="notes" linear="no"/>
</spine>
```

› 3.4.14 The guide Element [DEPRECATED]

The guide element [OPF2] is deprecated in favor of the landmarks feature in the EPUB Navigation Document. Refer to The landmarks nav Element [ContentDocs301] for more information.

Authors MAY include the guide element in the Package Document for EPUB 2 Reading System forwards compatibility purposes. EPUB 3 Reading Systems MUST ignore the guide element when provided in EPUB 3 Publications whose EPUB Navigation Document includes the landmarks feature.

› 3.4.15 The bindings Element

The bindings element defines a set of custom handlers for media types not supported by this specification.

Element Name

bindings

Usage

Optional fourth or fifth child of package , following spine or guide .

Attributes

None.

Content Model

One or more mediaType elements [required]

The package element MAY contain at most one bindings element.

The bindings element provides a means for Authors to include more sophisticated fallbacks than would otherwise be possible with the [HTML5] object element's intrinsic fallback mechanisms. When present, Reading Systems that support scripting MUST utilize the bindings element to handle object elements that reference unsupported media types.

Each of the bindings element's child mediaType elements defines a unique handler for one of the foreign media types referenced in the Rendition's XHTML Content Documents.

When an unsupported media type is encountered during processing of a document, the Reading System MUST look up the handler in the bindings element by checking the media-type attribute of each mediaType element for a match (and before attempting any other type of fallback processing). If a match is found, the XHTML Content Document referenced in the element's handler attribute MUST be instantiated instead of the referenced resource. If no match is found, the Reading System SHOULD continue with normal fallback processing (i.e., check for an intrinsic fallback for the object).

The Reading System MUST instantiate the designated handler as if it had been referenced from the object element's data attribute with the following parameters:

src

the value of which MUST be an IRI [RFC3987] to the resource (i.e., the value of the object element's data attribute).

type

the value of which MUST be the resource media type (i.e., the value of the object element's type attribute).

Any additional param children of the object element MUST be similarly added as parameters using the param element's name attribute as the new parameter name and its value attribute as the new value.

For example, the following object element containing a foreign media type:

```
<object data="horse.ogg" type="audio/ogg">
    <param name="autoplay" value="false"/>
</object>
```

would result in the following query string being sent to the handler XHTML Content Document after processing:

```
src=horse.ogg&type=audio/ogg&autoplay=false
```

All IRI reserved characters, plus the characters <, >, ", space, {, }, |, \, ^ and `, in the generated query string MUST be encoded and decoded as per [RFC3987].

object elements that reference media types handled by the bindings element are only processed in spine-referenced XHTML Content Documents (i.e., they are ignored in container-constrained scripting contexts).

› Example

The following partial example illustrates how bindings can be used to provide a slideshow.

Consider a Rendition of an EPUB Publication with the following Package Document:

```
<package …>
    …
    <manifest>
        <item id="pict1"
            href="images/Pict1.jpg"
            media-type="image/jpeg"/>
        …
        <item id="content"
            href="content.xhtml"
            media-type="application/xhtml+xml"/>
        <item id="impl"
```

```
            href="impl.xhtml"
            media-type="application/xhtml+xml"
            properties="scripted"/>
        <item id="slideshow"
            href="slideshow.xml"
            media-type="application/x-demo-slideshow"/>
    </manifest>

    <bindings>
        <mediaType handler="impl"
            media-type="application/x-demo-slideshow"/>
    </bindings>
    …
</package>
```

and the following content in the file `content.xhtml`:

```
<html …>
    …
    <body>
        …
        <object data="slideshow.xml"
            type="application/x-demo-slideshow">
            <img src="images/Pict1.jpg"/>
            <img src="images/Pict2.jpg"/>
            <img src="images/Pict3.jpg"/>
            <img src="images/Pict4.jpg"/>
        </object>
        …
    </body>
</html>
```

and the following content in the file `slideshow.xml`:

```
<slides>
    <slide src="images/Pict1.jpg" dur="3"/>
    <slide src="images/Pict2.jpg" dur="3"/>
    <slide src="images/Pict3.jpg" dur="3"/>
    <slide src="images/Pict4.jpg" dur="3"/>
</slides>
```

Depending on the capabilities of the User's Reading System, they will see one of the following renderings of the slideshow:

- If the Reading System supports the native slideshow format, it will render a rotating set of images as specified in `slideshow.xml`.

- If the Reading System cannot support the slideshow media type but supports scripting, it can check the `bindings` element in the Package Document for a scripted fallback. There it will find a reference to the `item` element containing the handler document (`impl.xhtml`). The Reading System can now load this document to render a JavaScripted equivalent of the slideshow (source not shown).

- If the Reading System does not support the slideshow media type and also does not support scripting, it will use the fallback images specified in the object element to show a static set of all the images.

› 3.4.16 The mediaType Element

The mediaType element associates a Foreign Resource media type with a handler XHTML Content Document.

Element Name

mediaType

Usage

As a child of bindings . Repeatable.

Attributes

media-type [required]

A media type [RFC2046] that specifies the type and format of the resource to be handled.

handler [required]

An IDREF [XML] that identifies the manifest XHTML Content Document to be invoked to handle content of the type specified in this element

Content Model

Empty

Each child mediaType of a bindings element MUST define a unique content type in its media-type attribute, and the media type specified MUST NOT be a Core Media Type.

The required handler attribute MUST reference the ID [XML] of an item in the manifest of the default implementation for this media type. The referenced item MUST be an XHTML Content Document.

All XHTML Content Documents designated as handlers MUST have the scripted property set in their manifest item element's properties attribute.

› 3.4.17 The collection Element

The collection element defines a related group of resources.

Element Name

collection

Usage

Optional sixth element of package. Repeatable.

Attributes

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) and rtl (right-to-left).

id [optional]

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

role [required]

Specifies the nature of the collection, as defined below.

Content Model

In this order: metadata [optional], (collection [1 or more] or (collection [0 or more], link [1 or more]))

The collection element allows resources to be assembled into logical groups for a variety of potential uses: enabling content that has been split across multiple EPUB Content Documents to be reassembled back into a meaningful unit (e.g., an index split across multiple documents), identifying resources for specialized purposes (e.g., preview content), or collecting together resources that present additional information about the given Rendition.

The collection element, as defined in this section, represents a generic framework from which specific implementations are intended to be derived (e.g., through IDPF sub-specifications). Such implementations MUST define the purpose of the collection element within a Rendition, as well as all requirements for its valid production and use (specifically any requirements that differ from the general framework presented below).

Each implementation MUST define a role value that uniquely identifies all conformant collection elements. The role of each collection element in the Package Document MUST be identified in its role attribute, whose value MUST be one or more NMTOKENs [XSD-DATATYPES] and/or full IRIs [RFC3987]. The use of NMTOKEN values is reserved for IDPF-defined roles, a registry of which is maintained at http://www.idpf.org/epub/vocab/package/roles . NMTOKEN values not defined in the registry are not valid. No roles are defined in this section.

Third parties MAY define custom roles for the collection element, but such roles MUST be identified using full IRIs. Custom roles MUST NOT incorporate the string idpf.org in the host component of their identifying IRI.

NOTE

To facilitate interoperability of custom roles across Reading Systems, implementers are strongly encouraged to document their use of the collection element at http://www.idpf.org/epub/extensions/roles .

The optional metadata element child of collection is an adaptation of the package metadata element, with the following differences in syntax and semantics:

- No metadata is required by default.
- Package-level restrictions on the use of metadata elements MAY be overridden.
- All primary metadata expressions apply to the collection.
- The refines attribute MUST NOT reference elements outside the containing collection.
- The OPF2 meta element MUST NOT be included.

A collection MAY define sub-collections through the inclusion of one or more child collection elements.

The link element child of collection is an adaptation of the metadata link element, with the following differences in syntax and semantics:

- The href attribute MAY reference any resource, including those listed in the manifest .
- The IRI value of the href attribute MAY have a fragment component to indicate that only a portion or subset of a resource is included in the collection.
- The rel attribute is OPTIONAL.
- The refines attribute MUST NOT be attached.

Each link MUST reference a resource that is a member of the group. The order of link elements is not significant.

Specific implementations of the collection element MAY tailor the requirements defined above to better reflect their needs (e.g., requiring metadata, imposing further restrictions on the use of elements and attributes, or making the order of link elements significant). However, the resulting content model MUST represent a valid subset of the one defined in this section (e.g., specific implementations cannot introduce new elements or attributes, or re-introduce those expressly forbidden above). Specific implementations MUST NOT define collections in a way that overrides the requirements of the manifest and spine .

In the context of this specification, support for collections in Reading Systems is OPTIONAL. Reading Systems MUST ignore collection elements that define unrecognized roles.

The rendering of a Rendition MUST NOT be dependent on the recognition of collection elements. The content MUST remain consumable by a User without any information loss or other significant deterioration.

› Examples

The following example shows the assembly of two XHTML Content Documents that represent a single unit.

```
<package …>
    …
    <collection role="http://example.org/roles/unit">
        <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
            <dc:title>Foo Bar</dc:title>
        </metadata>
        <link href="EPUB/xhtml/foo-1.xhtml"/>
        <link href="EPUB/xhtml/foo-2.xhtml"/>
    </collection>
    …
</package>
```

› 4 Package Metadata

› 4.1 Publication Identifiers

› 4.1.1 Unique Identifier

The Author is responsible for including a primary identifier in the Package Document metadata that is unique to one and only one EPUB Publication. This Unique Identifier, whether chosen or assigned, MUST be stored in a `dc:identifier` element and be referenced as the Unique Identifier in the `package` element `unique-identifier` attribute.

Although not static, changes to the Unique Identifier for an EPUB Publication SHOULD be made as infrequently as possible. New identifiers SHOULD NOT be issued when updating metadata, fixing errata or making other minor changes to the EPUB Publication.

› 4.1.2 Release Identifier

The Unique Identifier of an EPUB Publication typically SHOULD NOT change with each minor revision to the package or its contents, as Unique Identifiers are intended to have maximal persistence both for referencing and distribution purposes. Each release of an EPUB Publication normally requires that the new version be uniquely identifiable, however, which results in the contradictory need for reliable Unique Identifiers that are changeable.

To redress this problem of identifying minor modifications and releases without changing the Unique Identifier, this specification defines the semantics for a *Release Identifier*, or means of distinguishing and sequentially ordering EPUB Publications with the same Unique Identifier. The Release Identifier is not an actual property in the package `metadata` section, but is a value that can be obtained from two required pieces of metadata: the Unique Identifier and the last modification date of the Rendition.

When the taken together, the combined value represents a unique identity that can be used to distinguish any particular version of an EPUB Publication from another. To ensure that a Release Identifier can be constructed, each Rendition MUST include exactly one [DCTERMS] `modified` property containing its last modification date (see meta). The value of this property MUST be an XML Schema [XSD-DATATYPES] dateTime conformant date of the form:

```
CCYY-MM-DDThh:mm:ssZ
```

The last modification date MUST be expressed in Coordinated Universal Time (UTC) and MUST be terminated by the `Z` time zone indicator.

Although not a part of the package metadata, for referencing and other purposes all string representations of the identifier MUST be constructed using the at sign (`@`) as the separator (i.e., of the form "id`@`date"). Whitespace MUST NOT be included when concatenating the strings.

The following example shows how a Unique Identifier and modification date are combined to form the Release Identifier.

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

results in the Package ID:

urn:uuid:A1B0D67E-2E81-4DF5-9E67-A64CBE366809@2011-01-01T12:00:00Z
```

Note that it is possible that the separator character MAY occur in the Unique Identifier, as these identifiers MAY be any string value. The Release Identifier consequently MUST be split on the last instance of the at sign when decomposing it into its component parts.

The Release Identifier does not supersede the Unique Identifier, but represents the means by which different versions of the same EPUB Publication can be distinguished and identified in distribution channels and by Reading Systems. The sequential, chronological order inherent in the required format of the timestamp also places EPUB Publications in order without requiring knowledge of the exact identifier that came before.

The Release Identifier consequently allows a set of EPUB Publications to be inspected to determine if they represent the same version of the same Publication, different versions of a single EPUB Publication, or any combination of differing and similar EPUB Publications.

NOTE

When an EPUB Container includes more than one Rendition of an EPUB Publication, updating the last modified date of the Default Rendition for each release — even if it has not been updated — will help ensure that the EPUB Publication does not appear to be the same version as an earlier release, as Reading Systems are only required to process the Default Rendition.

› 4.2 Vocabulary Association Mechanisms

› 4.2.1 Overview

This section is informative

The property, properties, rel and scheme attributes use the property data type to represent terms from metadata vocabularies. Similar to a CURIE [RDFa11], the property data type represents an IRI [RFC3987] in compact form and simplifies the authoring of metadata from standardized vocabularies.

A property value is an expression that consists of a prefix and a reference, where the prefix — whether literal or implied — is a shorthand mapping of an IRI that typically resolves to a term vocabulary. When the prefix is converted to its IRI representation and combined with the reference, the resulting IRI normally resolves to a fragment within that vocabulary that contains human- and/or machine-readable information about the term.

To assist Reading Systems in processing property values, the means of establishing the IRI a prefix maps to is required, and this specification defines three such mechanisms:

- a default vocabulary — defines the mapping when a property value does not include a prefix;
- a set of reserved prefixes — these mappings are predefined (i.e., all Reading Systems recognize them) and can be used without having to be declared; and
- the prefix attribute — a declarative means of creating new prefix mappings on the root package element.

› 4.2.2 Default Vocabulary

The default vocabulary is a vocabulary that does not require a prefix to be declared in order to use its terms, and whose terms MUST always be unprefixed.

As the Package Document has multiple unrelated uses for metadata terms, a single default vocabulary is not defined. Instead, different default vocabularies are defined for use in attributes that accept a property data type as follows:

- The Package Metadata Vocabulary is defined to be the default vocabulary for the meta property, meta scheme, item properties and itemref properties attributes.

  If a property value in any of these attributes does not include a prefix, the IRI [RFC3987] stem `http://idpf.org/epub/vocab/package/#` MUST be used to generate the resulting IRI.

- The Link Relationships Vocabulary is defined to be the default vocabulary for the link rel attribute.

  If a property value in this attribute does not include a prefix, the IRI [RFC3987] stem `http://idpf.org/epub/vocab/package/link/#` MUST be used to generate the resulting IRI.

The IRIs associated with the Package Metadata Vocabulary and Link Relationships Vocabulary MUST NOT be assigned a prefix using the prefix attribute.

› 4.2.3 Reserved Prefixes

This specification reserves a set of prefixes that Authors MAY use in package metadata. These prefixes are defined in the normative document EPUB Package Document Reserved Prefixes.

The prefixes defined in this document are maintained and updated separately of this specification and are subject to change at any time.

Reading Systems MUST resolve all reserved prefixes used in Package Documents using their pre-defined URIs. Reserved prefixes SHOULD NOT be overridden in the prefix attribute, but Reading Systems MUST use such local overrides when encountered.

As changes to the reserved prefixes and updates to Reading Systems are not always going happen in synchrony, Reading Systems MUST NOT fail when encountering unrecognized prefixes (i.e., not reserved and not declared using the `prefix` attribute).

› 4.2.4 The `prefix` Attribute

The `prefix` attribute defines additional prefix mappings not reserved by the specification.

The value of the `prefix` attribute is a whitespace-separated list of one or more prefix-to-IRI mappings of the form:

(EBNF productions ISO/IEC 14977)
All terminal symbols are in the Unicode Block 'Basic Latin' (U+0000 to U+007F).

prefixes	=	mapping , { whitespace, { whitespace } , mapping } ;
mapping	=	prefix , ":" , space , { space } , ? xsd:anyURI ? ;
prefix	=	? xsd:NCName ? ;
space	=	#x20 ;
whitespace	=	(#x20 | #x9 | #xD | #xA) ;

The following example shows prefixes for the Friend of a Friend (`foaf`) and DBPedia (`dbp`) vocabularies being declared using the `prefix` attribute.

```
<package …
        prefix="foaf: http://xmlns.com/foaf/spec/
                 dbp: http://dbpedia.org/ontology/">
        …
</package>
```

To avoid conflicts, the `prefix` attribute MUST NOT be used to define a prefix that maps to the default vocabulary. If the `prefix` attribute includes a declaration for a pre-defined prefix, Reading Systems MUST use the URI mapping defined in the `prefix` attribute, regardless of whether of it maps to the same URI as the pre-defined prefix.

The prefix '_' is reserved for future compatibility with RDFa [RDFa11] processing, so MUST NOT be defined.

› 4.2.5 The property Data Type

› *4.2.5.1 Syntax*

The property data type is a compact means of expressing an IRI [RFC3987] and consists of an optional prefix separated from a reference by a colon.

(EBNF productions ISO/IEC 14977)
All terminal symbols are in the Unicode Block 'Basic Latin' (U+0000 to U+007F).

property	=	[prefix , ":"] , reference;	
prefix	=	? xsd:NCName ? ;	
reference	=	? irelative-ref ? ;	/* as defined in [RFC3987] */

The property data type is derived from the CURIE data type defined in [RDFa11], and represents a subset of CURIEs.

The following example shows a property value composed of the prefix `dcterms` and the reference `modified`.

```
<meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
```

After processing, this property would expand to the following IRI:

```
http://purl.org/dc/terms/modified
```

as the `dcterms:` prefix is a reserved prefix that maps to the IRI `http://purl.org/dc/terms/`.

When a prefix is omitted from the property value, the expressed reference represents a term from the default vocabulary.

The following example shows a property value taken from the default vocabulary.

```
<meta … property="role">aut</meta>
```

This property would expand to:

```
http://idpf.org/epub/vocab/package/#role
```

when the IRI for the default vocabulary is concatenated with the reference.

An empty string does not represent a valid property value, even though it is valid to the definition above.

› 4.2.5.2 Processing

A Reading System MUST use the following rules to create an IRI [RFC3987] from a property:

- If the property consists only of a reference, the IRI is obtained by concatenating the IRI stem associated with the default vocabulary to the reference.

- If the property consists of a prefix and reference, the IRI is obtained by concatenating the IRI stem associated with the prefix to the reference. If no matching prefix has been defined, the property is invalid and MUST be ignored.

The resulting IRI MUST be valid to [RFC3987]. Reading Systems are NOT REQUIRED to resolve this IRI, however.

› 4.3 Package Metadata Vocabulary

› 4.3.1 Overview

This section is informative

The following sections both define a set of properties for use in package metadata and constitute a referenceable vocabulary. This vocabulary is the default vocabulary reserved by this specification for the use of unprefixed terms in package metadata.

The properties defined in this vocabulary are referenceable using the base IRI `http://idpf.org/epub/vocab/package/#`.

> *NOTE*
>
> Property usage examples in the following sections have been drawn from the `metadata` and `meta` examples whenever possible. Refer to those examples for fuller context.

› 4.3.2 Metadata `meta` Properties

The `meta` element properties enhance Rendition metadata by providing additional level(s) of detail.

These properties MUST reference the expression or resource they augment in the `refines` attribute on their parent `meta` element.

The following tables detail the available properties.

› *4.3.2.1 Publication*

`alternate-script`	
Description:	The `alternate-script` property provides an alternate expression of the associated property value in a language and script identified by the `xml:lang` attribute. This property is typically attached to `creator` and `title` properties for internationalization purposes.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`

Extends: All properties.

Example:

```
<meta refines="#creator" property="alternate-script" xml:lang="ja">村上 春樹</meta>
```

belongs-to-collection

Description:

The belongs-to-collection property identifies the name of a collection to which the EPUB Publication belongs. An EPUB Publication MAY belong to one or more collections.

It is also possible chain these properties using the refines attribute to indicate that one collection is itself a member of another collection.

To allow Reading System to organize collections and avoid naming collisions (e.g., unrelated collections might share a similar name, or different editions of a collection could be released), an identifier SHOULD be provided that uniquely identifies the instance of the collection. The dcterms:identifier property must carry this identifier.

The collection MAY more precisely define its nature by attaching a collection-type property.

The position of the EPUB Publication within the collection MAY be provided by attaching a group-position property.

Allowed value(s): xsd:string

Cardinality:

In the metadata section: zero or more
Attached to other metadata: zero or more

Extends: Applies to the EPUB Publication, and can refine other instances of itself.

Example:

```
<meta property="belongs-to-collection" id="c01">The New French
Cuisine Masters</meta>
<meta refines="#c01" property="collection-type">series</meta>
<meta refines="#c01"
property="dcterms:identifier">urn:uuid:11111111-2222-3333-4444-
555555555555</meta>

<meta property="belongs-to-collection" id="c02">Harry
Potter</meta>
<meta refines="#c02" property="collection-type">set</meta>
<meta refines="#c02" property="group-position">2</meta>
<meta refines="#c02"
property="dcterms:identifier">urn:uuid:99999999-8888-7777-6666-
555555555555</meta>
```

collection-type

Description:

The collection-type property indicates the form or nature of a collection.

When the collection-type value is drawn from a code list or other formal enumeration, the scheme attribute SHOULD be attached to identify its source.

When a scheme is not specified, Reading Systems SHOULD recognize the following collection type values:

series

	A sequence of related works that are formally identified as a group; typically open-ended with works issued individually over time. `set` A finite collection of works that together constitute a single intellectual unit; typically issued together and able to be sold as a unit.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	`belongs-to-collection`

Example:	`<meta property="belongs-to-collection" id="c02">Harry Potter</meta>`
	`<meta refines="#c02" property="collection-type">set</meta>`

`display-seq`	
Description:	The `display-seq` property indicates the numeric position in which to display the current property relative to identical metadata properties (e.g., to indicate the order in which to render multiple `title`s). When the `display-seq` property is attached to some, but not all, of the members in a set, only the elements identified as having a sequence SHOULD be included in any rendering.
Allowed value(s):	`xsd:unsignedInt`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	All properties.
Example:	`<meta refines="#t2" property="display-seq">1</meta>`

`file-as`	
Description:	The `file-as` property provides the normalized form of the associated property for sorting.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	All properties.
Example:	`<meta refines="#creator" property="file-as">Murakami, Haruki</meta>`

`group-position`	
Description:	The `group-position` property indicates the numeric position in which the EPUB Publication is ordered relative to other works belonging to the same group (whether all EPUB Publications or not).

	The `group-position` property can be attached to any metadata property that establishes the group, but is typically associated with the `belongs-to-collection` property. An EPUB Publication can belong to more than one group.
Allowed value(s):	A single `xsd:unsignedInt` or series of decimal-separated numbers (e.g., `1` or `2.2.1`).
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	All properties.
Example:	`<meta refines="#c02" property="group-position">2</meta>`

`identifier-type`	
Description:	The `identifier-type` property indicates the form or nature of an `identifier`. When the `identifier-type` value is drawn from a code list or other formal enumeration, the `scheme` attribute SHOULD be attached to identify its source.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	`identifier`, `source`
Example:	`<meta refines="#src-id" property="identifier-type" scheme="onix:codelist5">15</meta>`

`meta-auth`	
Description:	The `meta-auth` property identifies the party or authority responsible for an instance of package metadata.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`
Extends:	All properties.
Example:	`<meta refines="isbn-id" property="meta-auth" id="meta-authority-01">http://isbn-international.org/</meta>`

`role`	
Description:	The `role` property describes the nature of work performed by a `creator` or `contributor` (e.g., that the person is the author or editor of a work). When the `role` value is drawn from a code list or other formal enumeration, the `scheme` attribute SHOULD be attached to identify its source.
Allowed value(s):	`xsd:string`
Cardinality:	In the metadata section: `zero or more` Attached to other metadata: `zero or one`

Extends:	contributor, creator
Example:	<meta refines="#creator" property="role" scheme="marc:relators">aut</meta>

source-of	
Description:	The source-of property indicates a unique aspect of an adapted source resource that has been retained in the given Rendition of the EPUB Publication. This specification defines the pagination value to indicate that the referenced source element is the source of the pagebreak properties defined in the content. This value SHOULD be set whenever pagination is included and the print source is known.
Allowed value(s):	pagination
Cardinality:	In the metadata section: zero or more Attached to other metadata: zero or one
Extends:	source
Example:	<meta refines="#isbn" property="source-of">pagination</meta>

title-type	
Description:	The title-type property indicates the form or nature of a title. When the title-type value is drawn from a code list or other formal enumeration, the scheme attribute SHOULD be attached to identify its source. When a scheme is not specified, Reading Systems SHOULD recognize the following title type values: main, subtitle, short, collection, edition and expanded.
Allowed value(s):	xsd:string
Cardinality:	In the metadata section: zero or more Attached to other metadata: zero or one
Extends:	title
Example:	<meta refines="#title" property="title-type">main</meta>

› 4.3.2.2 Rendering

rendition:flow	
Description:	Specifies the Author preference for how Reading Systems should handle content overflow.
Allowed value(s):	paginated | scrolled-continuous | scrolled-doc | auto The default value is auto
Cardinality:	Zero or one
Extends:	

	Sets the global value for the given Rendition. MUST NOT be set on a `meta` tag with a `refines` attribute.
Example:	`<meta property="rendition:flow">scrolled-doc</meta>`

`rendition:layout`	
Description:	Specifies whether the given Rendition is reflowable or pre-paginated.
Allowed value(s):	`reflowable` | `pre-paginated` The default value is `reflowable`
Cardinality:	Zero or one
Extends:	Sets the global value for the given Rendition. MUST NOT be set on a `meta` tag with a `refines` attribute.
Example:	`<meta property="rendition:layout">pre-paginated</meta>`

`rendition:orientation`	
Description:	Specifies which orientation the Author intends the given Rendition to be rendered in.
Allowed value(s):	`landscape` | `portrait` | `auto` The default value is `auto`
Cardinality:	Zero or one
Extends:	Sets the global value for the given Rendition. MUST NOT be set on a `meta` tag with a `refines` attribute.
Example:	`<meta property="rendition:orientation">landscape</meta>`

`rendition:spread`	
Description:	Specifies the intended Reading System synthetic spread behavior for the given Rendition.
Allowed value(s):	`none` | `landscape` | `portrait` | `both` | `auto` The default value is `auto`
Cardinality:	Zero or one
Extends:	Sets the global value for the given Rendition. MUST NOT be set on a `meta` tag with a `refines` attribute.
Example:	`<meta property="rendition:spread">both</meta>`

`rendition:viewport`	
Description:	Specifies the CSS initial containing block [CSS2.1] dimensions for pre-paginated XHTML and SVG Content Documents.
Required value:	`width=x, height=y`

Cardinality:	`Zero or one` without a `refines` attribute (global setting) `Zero or more` with a `refines` attribute (spine overrides)
Extends:	Applies globally for the given Rendition when no `refines` attribute set, otherwise applies to the `itemref` element referenced from the `refines` attribute.
Example:	`<meta property="rendition:viewport">width=1200, height=800</meta>`

› 4.3.3 Metadata `link` Properties

Properties for use in the `metadata` `link` element `rel` attribute are defined in the normative Link Relationships Vocabulary.

› 4.3.4 Manifest `item` Properties

The following tables define properties for use in the `manifest` `item` element `properties` attribute.

The **Applies to** field indicates which Publication Resource type(s) the given property MAY be specified on, the **Cardinality** field indicates the number of times the property MUST appear within the Package Document scope, and the **Usage** field indicates usage conditions.

`cover-image`	
Description:	The `cover-image` property identifies the described Publication Resource as the cover image for the Publication.
Applies to:	All raster and vector image types
Cardinality:	`Zero or one`
Usage:	Optional.

`mathml`	
Description:	The `mathml` property indicates that the described Publication Resource contains one or more instances of MathML markup.
Applies to:	EPUB Content Documents
Cardinality:	`Zero or more`
Usage:	MUST be set if and only if the criterion specified in **Description** above is met.

`nav`	
Description:	The `nav` property indicates that the described Publication Resource constitutes the EPUB Navigation Document of the given Rendition.
Applies to:	The EPUB Navigation Document
Cardinality:	`Exactly one`
Usage:	Required.

remote-resources	
Description:	The remote-resources property indicates that the described Publication Resource contains one or more internal references to other Publication Resources that are located outside of the EPUB Container. Refer to Publication Resource Locations for more information.
Applies to:	All Publication Resources with the capability of internal referencing (e.g., XHTML Content Documents, SVG Content Documents, EPUB Style Sheets and Media Overlay Documents).
Cardinality:	`Zero or more`
Usage:	MUST be set if and only if the criterion specified in **Description** above is met.

scripted	
Description:	The scripted property indicates that the described Publication Resource is a Scripted Content Document (i.e., contains scripted content and/or elements from HTML5 forms).
Applies to:	EPUB Content Documents
Cardinality:	`Zero or more`
Usage:	MUST be set if and only if the criterion specified in **Description** above is met.

svg	
Description:	The svg property indicates that the described Publication Resource embeds one or more instances of SVG markup. This property MUST be set when SVG markup is included directly in the resource and MAY be set when the SVG is referenced from the resource (e.g., from an [HTML5] img, object or iframe element).
Applies to:	XHTML Content Documents; the value is implied for SVG Content Documents.
Cardinality:	`Zero or more`
Usage:	MUST be set if and only if the criterion specified in **Description** above is met.

switch	
Description:	The switch property indicates that the described Publication Resource contains one or more instances of the epub:switch element.
Applies to:	XHTML Content Documents.
Cardinality:	`Zero or more`
Usage:	MUST be set if and only if the criterion specified in **Description** above is met.

The mathml, remote-resources, scripted and switch properties MUST be specified whenever the resource referenced by an item matches their respective definitions. These properties do not apply recursively to content included into a resource (e.g., via the HTML5 iframe element). For example, if a non-scripted XHTML Content Document embeds a scripted Content Document, only the embedded document's manifest item properties attribute will have the scripted value.

› Examples

The following example shows a manifest item *element that represents the EPUB Navigation Document.*

```
<item properties="nav" id="c1" href="c1.xhtml" media-
type="application/xhtml+xml" />
```

The following example shows a manifest item *element that represents the cover image.*

```
<item properties="cover-image" id="ci" href="cover.svg" media-
type="image/svg+xml" />
```

The following example shows a manifest item *element representing a Scripted Content Document that also contains embedded MathML.*

```
<item properties="scripted mathml" id="c2" href="c2.xhtml" media-
type="application/xhtml+xml" />
```

› 4.3.5 Spine itemref Properties

The following tables define properties for use in the itemref element properties attribute.

The **Cardinality** field indicates the number of times the property MUST appear within the Package Document scope, and the **Usage** field indicates usage conditions.

rendition:align-x-center	
Description:	Specifies that the given spine item should be centered horizontally in the viewport or spread.
Cardinality:	Zero or more
Usage:	Optional.

rendition:flow-auto	
Description:	Indicates no preference for overflow content handling by the Author.
Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:flow-paginated, rendition:flow-scrolled-continuous or rendition:flow-scrolled-doc properties.

rendition:flow-paginated	
Description:	Indicates the Author preference is to dynamically paginate content overflow.

Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:flow-auto, rendition:flow-scrolled-continuous or rendition:flow-scrolled properties.

rendition:flow-scrolled-continuous	
Description:	Indicates the Author preference is to provide a scrolled view for overflow content, and that consecutive spine items with this property are to be rendered as a continuous scroll. The scroll direction is defined relative to the block flow direction [CSS3WritingModes] of the root element of the XHTML Content Document referenced by the itemref element. The scroll direction is vertical if the block flow direction is downward (top-to-bottom). It is horizontal if the block flow direction of the root element is rightward (left-to-right) or leftward (right-to-left).
Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:flow-auto, rendition:flow-scrolled-doc or rendition:flow-paginated properties.

rendition:flow-scrolled-doc	
Description:	Indicates the Author preference is to provide a scrolled view for overflow content, and each spine item with this property is to be rendered as separate scrollable document.
Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:flow-auto, rendition:flow-scrolled-continuous or rendition:flow-paginated properties.

rendition:layout-pre-paginated	
Description:	Specifies that the given spine item is pre-paginated.
Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:layout-reflowable property.

rendition:layout-reflowable	
Description:	Specifies that the given spine item is reflowable.
Cardinality:	Zero or more
Usage:	Optional. This property MUST NOT be specified on an itemref that also specifies the rendition:layout-pre-paginated property.

rendition:orientation-auto	

Description:	Specifies that the Reading System can determine the orientation to rendered the spine item in.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:orientation-landscape` or `rendition:orientation-portrait` property.

`rendition:orientation-landscape`	
Description:	Specifies that the given spine item is to be rendered in landscape orientation.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:orientation-portrait` or `rendition:orientation-auto` property.

`rendition:orientation-portrait`	
Description:	Specifies that the given spine item is to be rendered in portrait orientation.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:orientation-landscape` or `rendition:orientation-auto` property.

`rendition:page-spread-center`	
Description:	Specifies the forced placement of a Content Document in a Synthetic Spread
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `page-spread-right` or `page-spread-left` properties.

`page-spread-left`	
Description:	The `page-spread-left` property indicates that the first page of the associated `item` element's EPUB Content Document represents the left-hand side of a two-page spread.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `page-spread-right` or `rendition:page-spread-center` properties.

`page-spread-right`	
Description:	The `page-spread-right` property indicates that the first page of the associated `item` element's EPUB Content Document represents the right-hand side of a two-page spread.
Cardinality:	`Zero or more`

Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `page-spread-left` or `rendition:page-spread-center` properties.

`rendition:spread-auto`	
Description:	Specifies the Reading System can determine when to render a synthetic spread for the spine item.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:spread-portrait`, `rendition:spread-landscape`, `rendition:spread-both` or `rendition:spread-none` property.

`rendition:spread-both`	
Description:	Specifies the Reading System should render a synthetic spread for the spine item in both portrait and landscape orientations.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:spread-portrait`, `rendition:spread-landscape`, `rendition:spread-auto` or `rendition:spread-none` property.

`rendition:spread-landscape`	
Description:	Specifies the Reading System should render a synthetic spread for the spine item only when in landscape orientation.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:spread-portrait`, `rendition:spread-both`, `rendition:spread-auto` or `rendition:spread-none` property.

`rendition:spread-none`	
Description:	Specifies the Reading System should not render a synthetic spread for the spine item.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:spread-portrait`, `rendition:spread-landscape`, `rendition:spread-both` or `rendition:spread-auto` property.

`rendition:spread-portrait`	
Description:	Specifies the Reading System should render a synthetic spread for the spine item only when in portrait orientation.
Cardinality:	`Zero or more`
Usage:	Optional. This property MUST NOT be specified on an `itemref` that also specifies the `rendition:spread-landscape`, `rendition:spread-both`,

	rendition:spread-auto or rendition:spread-none property.

› Examples

The following example shows how a two-page spread of a map might be indicated in the spine.

```
<spine>
        <itemref idref="title"/>
        <itemref idref="ps-1-l" properties="page-spread-left"/>
        <itemref idref="ps-1-r" properties="page-spread-right"/>
        <itemref idref="toc"/>
        …
</spine>
```

› 4.4 Publication Rendering

› 4.4.1 General Properties

› *4.4.1.1 Overview*

This section is informative

Not all rendering information can be expressed through the underlying technologies that EPUB is built upon. Although XHTML with CSS provides powerful layout capabilities, for example, those capabilities are limited to the scope of the document being rendered.

This section defines general-purpose properties that allow Authors to express package-level rendering intentions (i.e., functionality that can only be implemented by the EPUB Reading System). If a Reading System supports the desired rendering, these properties enable the User to be presented the content as the Author optimally designed it.

› *4.4.1.2 The* rendition:flow *Property*

› *4.4.1.2.1 Usage*

When the rendition:flow property is specified on a meta element, it indicates the Author's global preference for overflow content handling (i.e., for all spine items). Authors MAY indicate a preference for dynamic pagination or scrolling. For scrolled content, it is also possible to specify whether consecutive EPUB Content Documents are to be rendered as a continuous scrolling view or whether each is to be rendered separately (i.e., with a dynamic page break between each).

If a Reading System supports the specified rendering, it SHOULD use that method to handle overflow content, but MAY provide the option for Users to override the requested rendering.

The default value auto MUST be assumed by Reading Systems as the global value if no meta element carrying this property occurs in the metadata section. Reading Systems MAY support only this default value.

If a Reading Systems supports the `rendition:layout` property, it MUST ignore the `rendition:flow` property when it has been set on a spine item that also specifies the `rendition:layout` value `pre-paginated`.

› 4.4.1.2.2 Allowed values

The following values are defined for use with the `rendition:flow` property:

paginated

> The Reading System SHOULD dynamically paginate all overflow content.

scrolled-continuous

> The Reading System SHOULD render all Content Documents such that overflow content is scrollable, and the EPUB Publication represented by the given Rendition SHOULD be presented as one continuous scroll from spine item to spine item (except where locally overridden).

scrolled-doc

> The Reading System SHOULD render all Content Documents such that overflow content is scrollable, and each spine item SHOULD be presented as a separate scrollable document.

auto

> The Author does not have a preference for overflow handling. The Reading System MAY render overflow content using its default method or a User preference, whichever is applicable.

› Examples

The following example demonstrates an Author's intent to have a paginated Rendition with a scrollable table of contents.

```
<metadata>
    <meta property="rendition:flow">paginated</meta>
</metadata>

<spine>
    <itemref idref="toc" properties="rendition:flow-scrolled-doc"/>
    <itemref idref="c01"/>
</spine>
```

› 4.4.1.2.3 Spine Overrides

The `rendition:flow-auto` , `rendition:flow-paginated` , `rendition:flow-scrolled-continuous` , `rendition:flow-scrolled-doc` and properties MAY be specified locally on spine `itemref` elements, and will, in such cases, override the global value for the given spine item.

› 4.4.1.3 The `rendition:align-x-center` Property

When the rendition:align-x-center property is set on a spine item, it indicates that the rendered content SHOULD be centered horizontally within the viewport or spread, as applicable. This property does not affect the rendering of the spine item, only the placement of the resulting content box.

For reflowable content, Reading Systems that support this property MUST center each virtual page.

This version of this specification does not define a default rendering behavior when this property is not supported or specified. Reading Systems MAY render spine items by their own design.

NOTE

This property was developed primarily to handle "Naka-Tobira (中扉)" (sectional title pages), in the absence of reliable centering control within the content rendering. As support for paged media evolves in CSS, however, this property may become obsolete. Authors are encouraged to use CSS solutions when effective.

› 4.4.2 Fixed-Layout Properties

› *4.4.2.1 Overview*

This section is informative

EPUB documents, unlike print books or PDF files, are designed to change. The content flows, or reflows, to fit the screen and to fit the needs of the User. As noted in Rendering and CSS [EPUB3Overview] “content presentation should adapt to the User rather than the User having to adapt to a particular representation of content.”

But this principle doesn’t work for all types of documents. Sometimes content and design are so intertwined they cannot be separated. Any change in appearance risks changing the meaning, or losing all meaning. Fixed-Layout Documents give Authors greater control over presentation when a reflowable EPUB is not suitable for the content.

This section defines a set of metadata properties to allow declarative expression of intended rendering behaviors of Fixed-Layout Documents in the context of EPUB 3.

NOTE

EPUB 3 affords multiple mechanisms for representing fixed-layout content. When fixed-layout content is necessary, the Author's choice of mechanism will depend on many factors including desired degree of precision, file size, accessibility, etc. This section does not attempt to dictate the Author's choice of mechanism.

› *4.4.2.2 The `rendition:layout` Property*

› *4.4.2.2.1 Usage*

When the `rendition:layout` property is specified on a `meta` element, it indicates that the paginated or reflowable layout style applies globally for the EPUB Publication (i.e., for all spine items).

The default value `reflowable` MUST be assumed by EPUB Reading Systems as the global value if no `meta` element carrying this property occurs in the `metadata` section.

When the `rendition:layout` property is set to `pre-paginated`, Reading Systems MUST NOT include space between the adjacent content slots when rendering Synthetic Spreads.

When the property is set to `pre-paginated` for a spine item, its content dimensions MUST be set as defined in Fixed-Layout Documents [ContentDocs301] .

> *NOTE*
>
> Refer to rendition:viewport property for how to additionally declare the dimensions within the package metadata to facilitate Reading System optimization of the rendering.

› *4.4.2.2.2 Allowed values*

The following values are defined for use with the `rendition:layout` property:

reflowable

The given Rendition is not pre-paginated. Reading Systems MAY apply dynamic pagination when rendering.

pre-paginated

The given Rendition is pre-paginated. Reading Systems MUST produce exactly one page per spine `itemref` when rendering.

> *NOTE*
>
> Reading Systems typically restrict or deny the application of User or User Agent style sheets to pre-paginated documents, since, as a result of intrinsic properties of such documents, dynamic style changes are highly likely to have unintended consequences. Authors should take into account the negative impact on usability and accessibility that these restrictions have when choosing to use pre-paginated instead of reflowable content. Refer to Guideline 1.4 - Provide text configuration of the W3C User Agent Accessibility Guidelines for related information.

› Examples

The following example demonstrates fully fixed-layout content, using [MediaQueries] to apply different style sheets for three different device categories.

› Package Document

```
<meta property="rendition:layout">pre-paginated</meta>
```

› XHTML

```
<head>
    <meta name="viewport" content="width=1200, height=900"/>
    <link rel="stylesheet" href="eink-style.css" media="(max-monochrome:
3)"/>
    <link rel="stylesheet" href="skinnytablet-style.css" media="((color)
and
        (max-height:600px) and (orientation:landscape), (color) and
(max-width:600px)
        and (orientation:portrait))"/>
    <link rel="stylesheet" href="fattablet-style.css" media="((color)
and
        (min-height:601px) and (orientation:landscape), (color) and
(min-width:601px)
        and (orientation:portrait)"/>
</head>
```

Note that the Media Queries only affect the style sheet applied to the document. The size of the content area set in the `viewport` `meta` tag is static.

› 4.4.2.2.3 Spine Overrides

The `rendition:layout-pre-paginated` and `rendition:layout-reflowable` properties MAY be specified locally on spine `itemref` elements, and will, in such cases, override the global value for the given spine item.

› 4.4.2.3 The `rendition:orientation` property

› 4.4.2.3.1 Usage

When the `rendition:orientation` property is specified on a `meta` element, it indicates that the intended orientation applies globally for the given Rendition (i.e., for all spine items).

The default value `auto` MUST be assumed by Reading Systems as the global value if no `meta` element carrying this property occurs in the `metadata` section.

› 4.4.2.3.2 Allowed values

The following values are defined for use with the `rendition:orientation` property:

landscape

The given Rendition is intended for landscape rendering.

portrait

The given Rendition is intended for portrait rendering.

auto

The given Rendition is not orientation constrained.