TECHNICAL SPECIFICATION

ISO/IEC TS 30135-5

First edition
2014-11-15

Information technology — Digital publishing — EPUB3 —

Part 5: Media Overlay

Technologies de l'information — Publications numériques — EPUB3 —

Partie 5: Superposition de médias

Reference number
ISO/IEC TS 30135-5:2014(E)

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

Published in Switzerland

Foreword

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

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

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

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

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order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn.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again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

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

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

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

- *Part 1: Overview*
- *Part 2: Publications*
- *Part 3: Content Documents*
- *Part 4: Open Container Format*
- *Part 5: Media Overlay*
- *Part 6: Canonical Fragment Identifier*
- *Part 7: Fixed-Layout Documents*

EPUB Media Overlays 3.0

Recommended Specification 11 October 2011

THIS VERSION

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

LATEST VERSION

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

PREVIOUS VERSION

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

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

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

Editors

Marisa DeMeglio, DAISY Consortium

Daniel Weck, DAISY Consortium

TABLE OF CONTENTS

1 Overview

1.1 Purpose and Scope

This section is informative

This specification, EPUB Media Overlays 3.0, defines a usage of [SMIL] (Synchronized Multimedia Integration Language), the Package Document, the EPUB® Style Sheet, and the EPUB Content Document for representation of audio synchronized with the EPUB Content Document.

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 Publications 3.0 [Publications30], which defines publication-level semantics and overarching conformance requirements for EPUB Publications.
- EPUB Content Documents 3.0 [ContentDocs30], which defines profiles of XHTML, SVG and CSS for use in the context of EPUB Publications.
- EPUB Open Container Format (OCF) 3.0 [OCF3], which defines a file format and processing model for encapsulating a set of related resources into a single-file (ZIP) EPUB Container.

1.2 Relationship to Other Specifications

This section is informative

This specification relies on a subset of [SMIL], from which the EPUB Media Overlays elements and attributes defined in Media Overlay Document Definition are derived.

1.3 Terminology

EPUB Publication (or Publication)

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

Publication Resource

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

With the exception of the Package Document itself, Publication Resources must be listed in the manifest [Publications30] and must be bundled in the EPUB container file unless specified otherwise in Publication Resource Locations [Publications30].

Examples of resources that are not Publication Resources include those identified by the Package Document link [Publications30] element and those identified in outbound hyperlinks that resolve outside the EPUB Container (e.g., referenced from an [HTML5] `a` element `href` attribute).

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 [Publications30].

XHTML Content Document

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

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

SVG Content Document

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

EPUB Navigation Document

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

Core Media Type

A set of Publication Resource types for which no fallback is required. Refer to Publication Resources [Publications30] for more information.

Package Document

A Publication Resource carrying bibliographical and structural metadata about the EPUB

Publication, as defined in Package Documents [Publications30].

Manifest

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

Refer to manifest [Publications30] for more information.

Spine

An ordered list of Publication Resources, typically EPUB Content Documents, representing the default reading order of the Publication.

Refer to spine [Publications30] 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 this specification.

Text-to-Speech (TTS)

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

EPUB Style Sheet (or Style Sheet)

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

Viewport

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

CSS Viewport

A Viewport capable of displaying CSS-styled content.

EPUB Container (or Container)

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

Author

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

User

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

EPUB Reading System (or Reading System)

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

› 1.4 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 Media Overlay Document Definition

2.1 Introduction

This section is informative

Books featuring synchronized audio narration are found in mainstream e-books, educational tools and e-books formatted for persons with print disabilities. In EPUB 3, these types of books are created by using Media Overlay Documents to describe the timing for the pre-recorded audio narration and how it relates to the EPUB Content Document markup. The file format for Media Overlays is defined as a subset of SMIL, a W3C recommendation for representing synchronized multimedia information in XML.

The Media Overlays feature is designed to be transparent to EPUB Reading Systems that do not support the feature. The inclusion of Media Overlays in an EPUB Publication has no impact on the ability of Media Overlay-unaware Reading Systems to render that Publication as a "regular" EPUB Publication.

Although future versions of this specification may incorporate support for video media (e.g., synchronized text/sign-language books), this version supports only synchronizing audio media with the EPUB Content Document.

2.2 Content Conformance

A Media Overlay Document must meet all of the following criteria:

Document Properties

- It must meet the conformance constraints for XML documents defined in XML Conformance [Publications30].
- It must be valid to the Media Overlays schema as defined in Appendix A, *Media Overlays Schema* and conform to all content conformance constraints expressed in Media Overlay Document Definition.
- It must be authored to reflect the structure of the EPUB Content Document with which it is associated, as stated in Structure .
- Authors should avoid using scripts to control audio and video embedded in the EPUB Content Document, as stated in Embedded Audio and Video.
- It should use semantic markup where appropriate, as described in Semantic Inflection.
- It must be packaged with the EPUB Publication as shown in Packaging.

File Properties

- The Media Overlay Document filename should use the file extension `.smil`.

2.3 Reading System Conformance

EPUB Reading System support for Media Overlays is optional. A Reading System that supports Media Overlays must meet the following criteria:

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

› It must support XHTML Content Documents, and it may support SVG Content Documents.

› It must render Media Overlay elements as described in Basic Playback.

› It must allow User navigation while a Media Overlay is being played, as discussed in Navigation.

› It must adhere to rules regarding referenced audio and video embedded in the EPUB Content Document, as stated in Embedded Audio and Video.

› Text-to-Speech (TTS)-capable Reading Systems should conform to Reading System Text-to-Speech Conformance Requirements [Publications30].

› It should offer the skippability and escapability features described in Skippability and Escapability.

A Reading System that does not support Media Overlays must meet the following criteria:

› It must ignore both the `media-overlay` attribute on manifest `item` elements and the manifest `item` elements where the `media-type` attribute value equals `application/smil+xml`.

› 2.4 Media Overlay Document Definition

All elements [XML] defined in this section are in the `http://www.w3.org/ns/SMIL` namespace [XMLNS] unless otherwise specified.

› 2.4.1 The `smil` Element

The `smil` element must be the root element of all Media Overlay Documents.

Element Name

`smil`

Usage

The `smil` element is the root element of the Media Overlay Document.

Attributes

`version` [required]

Specifies the version number of the [SMIL] specification to which the Media Overlay adheres.

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

`id` [optional]

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

`epub:prefix` [optional]

Declares additional metadata vocabulary prefixes.

Refer to Semantic Inflection for more information.

Content Model

In this order: head [optional], body [required]

2.4.2 The head Element

The head element is the container for metadata in the Media Overlay Document, and consists of zero or one child metadata element.

Element Name

head

Usage

The head element is the optional first child of the smil element.

Attributes

None.

Content Model

metadata [0 or 1].

As this specification defines no metadata properties that must occur in the Media Overlay Document, the head element is optional.

2.4.3 The metadata Element

The metadata element represents metadata for the Media Overlay Document. The metadata element is an extension point that allows the inclusion of metadata from any metainformation structuring language.

Element Name

metadata

Usage

As a child of the head element.

Attributes

None.

Content Model

[0 or more] elements from any namespace.

This specification defines no metadata properties that must occur in the Media Overlay Document; the metadata element is provided for custom metadata requirements.

› 2.4.4 The body Element

The body element is the starting point for the presentation contained in the Media Overlay Document. It contains the main sequence of par and seq elements.

Element Name

body

Usage

The body element is the required second child of the smil element.

Attributes

epub:type [optional]

An expression of the structural semantics of the corresponding element in the EPUB Content Document.

The value is a whitespace separated list of property [Publications30] types. Refer to Semantic Inflection for more information.

id [optional]

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

epub:textref [optional]

The relative IRI reference [RFC3987] of the corresponding EPUB Content Document, including a fragment identifier that references the specific element as per the [XPTRSH].

Content Model

In any order: seq [0 or more] or par [0 or more]

At least one par or seq is required.

› 2.4.5 The seq Element

The seq element contains media objects which are to be rendered sequentially.

Element Name

seq

Usage

One or more seq elements may occur as children of the body element and of the seq element.

Attributes

epub:type [optional]

An expression of the structural semantics of the corresponding element in the EPUB Content Document.

The value is a whitespace separated list of property [Publications30] types. Refer to Semantic Inflection for more information.

`id` [optional]

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

`epub:textref` [required]

The relative IRI reference [RFC3987] of the corresponding EPUB Content Document, including a fragment identifier that references the specific element as per the [XPTRSH].

Content Model

In any order: `seq` [0 or more] or `par` [0 or more].

At least one `par` or `seq` is required.

› 2.4.6 The `par` Element

The `par` element contains media objects which are to be rendered in parallel.

Element Name

`par`

Usage

One or more `par` elements may occur as children of the body and seq elements.

Attributes

`epub:type` [optional]

An expression of the structural semantics of the corresponding element in the EPUB Content Document.

The value is a whitespace separated list of property [Publications30] types. Refer to Semantic Inflection for more information.

`id` [optional]

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

Content Model

In any order: `text` [required] and `audio` [optional]

The audio element is optional only if its sibling text element refers to audio or video media (see Embedded Audio and Video), or to textual content intended for rendering via Text-to-Speech (TTS).

› 2.4.7 The `text` Element

The `text` element references an element in the EPUB Content Document. A `text` element typically refers to a textual element, but can also refer to other EPUB Content Document media elements (see

Embedded Audio and Video).

Element Name

`text`

Usage

As a required child of the par element.

Attributes

`src` [required]

The relative IRI reference [RFC3987] of the corresponding EPUB Content Document, including a fragment identifier that references the specific element as per the [XPTRSH].

`id` [optional]

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

Content Model

Empty.

› 2.4.8 The `audio` Element

The `audio` element represents a clip of audio media.

Element Name

`audio`

Usage

A required child of the par element unless its sibling text element refers to audio or video media, in which case it is optional (see Embedded Audio and Video).

Attributes

`id` [optional]

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

`src` [required]

The relative or absolute IRI reference [RFC3987] of an audio file. The audio file must be one of the audio formats listed in the Core Media Types [Publications30] table.

`clipBegin` [optional]

A clock value that specifies the offset into the physical media corresponding to the start point of an audio clip.

Clock values are a subset of SMIL clock values, defined in [SMIL]. See Appendix B, *Examples of Clock Values*.

`clipEnd` [optional]

A clock value that specifies the offset into the physical media corresponding to the

end point of an audio clip.

Clock values are a subset of SMIL clock values, defined in [SMIL]. See Appendix B, *Examples of Clock Values*.

The chronological offset of the terminating position must be after the starting offset specified in the `clipBegin` attribute.

Content Model

Empty.

3 Creating Media Overlays

3.1 Overview

This section is informative

A pre-recorded narration of a publication can be represented as a series of audio clips, each corresponding to part of the EPUB Content Document. A single audio clip, for example, typically represents a single phrase or paragraph, but infers no order relative to the other clips or to the text of a document. Media Overlays solve this problem of synchronization by tying the structured audio narration to its corresponding text (or other media) in the EPUB Content Document using SMIL markup. Media Overlays are, in fact, a simplified subset of SMIL 3.0 that allow the playback sequence of these clips to be defined.

The SMIL elements primarily used for structuring Media Overlays are `body` (used for the main sequence), `seq` (sequence) and `par` (parallel). (Refer to Media Overlay Document Definition for more information on these and other SMIL elements.)

The `par` element is the basic building block of an Overlay and corresponds to a phrase in the EPUB Content Document. The element provides two key pieces of information for synchronizing content: 1) the audio clip containing the narration for the phrase; and 2) a pointer to the associated EPUB Content Document fragment. The `par` element uses two media element children to represent this information: an `audio` element and a `text` element. Since `par` elements render their children in parallel, the audio clip and EPUB Content Document fragment are played at the same time, resulting in a synchronized presentation.

The `text` element `src` attribute references the associated phrase, sentence, or other segment of the EPUB Content Document by its IRI reference. The `audio` element `src` attribute similarly references the location of the corresponding audio clip, and adds the optional `clipBegin` and `clipEnd` attributes to indicate a specific offset within the clip.

The following example shows the Media Overlays markup for a single phrase or sentence.

```
<par>
    <text src="chapter1.xhtml#sentence1"/>
    <audio src="chapter1_audio.mp3" clipBegin="23s" clipEnd="30s"/>
</par>
```

`par` elements are placed together sequentially to form a series of phrases or sentences. Not every element of the EPUB Content Document will have a corresponding `par` element in the Media Overlay, only those relevant to the audio narration.

The following example shows a basic Media Overlay Document containing a sequence of phrases. The `body` *element acts as the main sequence for the whole document.*

```
<smil xmlns="http://www.w3.org/ns/SMIL"
      version="3.0">
    <body>
        <par id="par1">
            <text src="chapter1.xhtml#sentence1"/>
            <audio src="chapter1_audio.mp3" clipBegin="0s"
clipEnd="10s"/>
        </par>
        <par id="par2">
            <text src="chapter1.xhtml#sentence2"/>
            <audio src="chapter1_audio.mp3" clipBegin="10s"
clipEnd="20s"/>
        </par>
        <par id="par3">
            <text src="chapter1.xhtml#sentence3"/>
            <audio src="chapter1_audio.mp3" clipBegin="20s"
clipEnd="30s"/>
        </par>
    </body>
</smil>
```

`par` elements can also be added to `seq` elements to define more complex structures such as parts and chapters (see Structure).

› 3.2 Relationship to the EPUB Content Document

NOTE

In this section, the EPUB Content Document is assumed to be an XHTML Content Document. While Media Overlays can be used with SVG Content Documents, playback behavior might not be consistent and therefore interoperability is not guaranteed.

› 3.2.1 Structure

The ordering of the Media Overlay elements must match the default reading order of the EPUB Content Document. The `par` element represents phrases, and the `seq` element (sequence) represents nested EPUB Content Document containers such as sections, asides, headers, and footnotes. `seq` children must be other `seq` or `par` elements. Each `seq` element must contain an `epub:textref` attribute which references the corresponding EPUB Content Document element by IRI reference.

The following example shows a Media Overlay Document with nested `seq` *elements, representing a chapter with both a section header and a sidebar, which itself has a nested figure.*

```
<smil xmlns="http://www.w3.org/ns/SMIL"
      xmlns:epub="http://www.idpf.org/2007/ops"
      version="3.0">
    <body>

        <!-- a chapter -->
        <seq id="id1" epub:textref="chapter1.xhtml#sectionstart"
epub:type="chapter">
```

```
            <!-- the section title -->
            <par id="id2">
                <text src="chapter1.xhtml#section1_title"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:23:23.84"
clipEnd="0:23:34.221"/>
            </par>

            <!-- some sentences in the chapter -->
            <par id="id3">
                <text src="chapter1.xhtml#text1"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:23:34.221"
clipEnd="0:23:59.003"/>
            </par>
            <par id="id4">
                <text src="chapter1.xhtml#text2"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:23:59.003"
clipEnd="0:24:15.000"/>
            </par>

            <!-- an informational sidebar -->
            <seq id="id5" epub:textref="chapter1.xhtml#sidebar"
epub:type="sidebar">
                <par id="id6">
                    <text src="chapter1.xhtml#sidebartitle"/>
                    <audio src="chapter1_audio.mp3"
clipBegin="0:24:15.000" clipEnd="0:24:18.123"/>
                </par>

                <!-- a figure within the sidebar -->
                <seq id="id7" epub:textref="chapter1.xhtml#figure">
                    <par id="id8">
                        <text src="chapter1.xhtml#photo"/>
                        <audio src="chapter1_audio.mp3"
clipBegin="0:24:18.123" clipEnd="0:24:28.764"/>
                    </par>
                    <par id="id9">
                        <text src="chapter1.xhtml#caption"/>
                        <audio src="chapter1_audio.mp3"
clipBegin="0:24:28.764" clipEnd="0:24:50.010"/>
                    </par>
                </seq>

                <!-- some sentences in the sidebar -->
                <par id="id10">
                    <text src="chapter1.xhtml#sidebartext1"/>
                    <audio src="chapter1_audio.mp3"
clipBegin="0:24:50.010" clipEnd="0:25:28.530"/>
                </par>
                <par id="id11">
                    <text src="chapter1.xhtml#sidebartext2"/>
                    <audio src="chapter1_audio.mp3"
clipBegin="0:25:28.530" clipEnd="0:25:45.515"/>
                </par>
            </seq>

            <!-- more sentences in the chapter (outside the sidebar) -->
            <par id="id12">
                <text src="chapter1.xhtml#text3"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:25:45.515"
clipEnd="0:26:30.203"/>
```

```
            </par>
            <par id="id13">
                <text src="chapter1.xhtml#text4"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:26:30.203"
clipEnd="0:27:15.000"/>
            </par>

        </seq>
    </body>
</smil>
```

The reason for grouping structures like sidebars, section headers, figures, tables, and footnotes in a `seq` element is so that their start and end positions can be identified during playback. Reading Systems can then offer playback options tailored to the layout of the Publication, such as jumping past a long sidebar, turning off rendering of page break announcements (see Skippability and Escapability), or customizing the reading mode to suit structures such as tables.

The following example shows the EPUB Content Document that corresponds to the previous Media Overlay example.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:epub="http://www.idpf.org/2007/ops"
      xml:lang="en"
      lang="en">
    <head>
        <title>Media Overlays Example of EPUB Content Document</title>
    </head>
    <body id="sec1">
        <section id="sectionstart" epub:type="chapter">
            <h1 id="section1_title">The Section Title</h1>
            <p id="text1">The first phrase of the main text body.</p>
            <p id="text2">The second phrase of the main text body.</p>
            <aside id="sidebar" epub:type="sidebar">
                <h2 id="sidebartitle">The Sidebar Title</h2>
                <figure id="figure">
                    <img id="photo"
                         src="photo.png"
                         alt="a photograph for which there is a caption"
/>
                    <figcaption id="caption">The photo
caption</figcaption>
                </figure>
                <p id="sidebartext1">A phrase in the sidebar.</p>
                <p id="sidebartext2">Another phrase in the sidebar</p>
            </aside>
            <p id="text3">The third phrase of the main text body.</p>
            <p id="text4">The fourth phrase of the main text body.</p>
        </section>
    </body>
</html>
```


› 3.2.2 Granularity

This section is informative

Media Overlay `text` elements' `src` attributes refer to EPUB Content Document elements by their IDs [XML]. The granularity level of the Media Overlay therefore depends on how the EPUB Content Document

is marked up. If the finest level of markup is at the paragraph level, then that is the finest possible level at which Media Overlay synchronization can be authored. Likewise, if sub-paragraph markup is available, such as [HTML5] `span` elements representing phrases or sentences, then finer granularity is possible in the Media Overlay. Finer granularity gives Users more precise results for synchronized playback when navigating by word or phrase and when searching the text, but increases the file size of the Media Overlay Documents.

› 3.2.3 Embedded Audio and Video

Any EPUB Content Document associated with a Media Overlay may contain embedded media such as video, audio, and images. The Media Overlay `text` element may be used in such instances to reference the embedded media by its ID [XML] value.

When a `text` element references embedded media that contains audio, no `audio` sibling element is required, though one is allowed.

Authors should avoid using scripts to control playback of referenced embedded EPUB Content Document media, as this may conflict with Media Overlays playback behavior.

› 3.2.4 Text-to-Speech

This specification allows the use of Text-to-Speech (TTS) in addition to pre-recorded audio clips. When a Media Overlay `text` element with no `audio` sibling element references an element within the target EPUB Content Document, the contents of that referenced element must be appropriate for rendering via TTS. For example, it could be a textual EPUB Content Document element or contain a text fallback.

› 3.3 Semantic Inflection

In order to express semantic inflections, the epub:type attribute [ContentDocs30] may be attached to Media Overlay `par`, `seq`, and `body` elements.

Values for the Media Overlay `epub:type` attribute are constrained identically to the `epub:type` attribute in EPUB Content Document. Refer to Semantic Inflection [ContentDocs30] for details.

The `epub:type` attribute facilitates Reading System behavior appropriate for the semantic type(s) indicated. Examples of these behaviors are Skippability and Escapability and Table Reading Mode.

The following example shows the semantic markup for a Media Overlay containing a sidebar.

```
<smil xmlns="http://www.w3.org/ns/SMIL"
      xmlns:epub="http://www.idpf.org/2007/ops"
      version="3.0">
    <body>
        <seq id="id1" epub:textref="chapter1.xhtml#sidebar"
epub:type="sidebar">
            <par id="id2">
                <text src="chapter1.xhtml#sidebartitle"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:24:15.000"
clipEnd="0:24:18.123"/>
            </par>
            <par id="id3">
                <text src="chapter1.xhtml#sidebartext1"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:24:18.123"
clipEnd="0:24:38.530"/>
            </par>
            <par id="id4">
```

```
                <text src="chapter1.xhtml#sidebartext2"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:24:38.530"
clipEnd="0:25:00.515"/>
            </par>
        </seq>
    </body>
</smil>
```

This specification adopts the vocabulary association mechanisms defined in Vocabulary Association [ContentDocs30] unmodified. Terms from the default vocabulary [ContentDocs30] must be used unprefixed in Overlay Documents.

› 3.4 Associating Style Information

Visual rendering information for the currently-playing EPUB Content Document element may be expressed in the EPUB Style Sheet using an author-defined class. This author-defined class name should be declared in the Package Document metadata, using the metadata property `active-class`. The class name is then discoverable by Reading Systems.

This example demonstrates how authors may associate style information with the currently-playing EPUB Content Document element.

> NOTE
>
> Although this example uses the class name `-epub-media-overlay-active`, any class name is permitted.

The author-defined CSS class name, declared using the metadata property active-class in the Package Document:

```
<meta property="media:active-class">-epub-media-overlay-active</meta>
```

The EPUB Style Sheet containing the author-defined class name:

```
.-epub-media-overlay-active
{
    background-color: yellow;
}
```

The relevant EPUB Content Document excerpt:

```
<span id="txt1">This is the first phrase.</span>
<span id="txt2">This is the second phrase.</span>
<span id="txt3">This is the third phrase.</span>
```

In this example, the Reading System would apply the author-defined `-epub-media-overlay-active` class to each text element in the EPUB Content Document as it became active during playback. Conversely, the class name is removed when the element is no longer active. The User would see each EPUB Content Document element styled with a yellow background for the duration of that element's playback.

› 3.5 Packaging

› 3.5.1 Including Media Overlays

Manifest item elements [Publications30] in the Package Document may specify a Media Overlay via the media-overlay attribute. Media Overlays are themselves manifest items and must be referred to by their IDs [XML].

The following example shows how to include Media Overlays in the manifest of a Package Document.

```
<manifest>
    <item id="ch1"
          href="chapter1.xhtml"
          media-type="application/xhtml+xml"
          media-overlay="ch1_audio"/>
    <item id="ch1_audio"
          href="chapter1_audio.smil"
          media-type="application/smil+xml"/>
</manifest>
```

Manifest items which refer to Media Overlays must have the media-type `application/smil+xml` as specified in Core Media Types [Publications30].

The `media-overlay` attribute must be attached to manifest `item`s that reference EPUB Content Documents only.

A single Media Overlay file may refer to more than one EPUB Content Document, but an EPUB Content Document must not be referenced by more than one Media Overlay file.

Not every EPUB Content Document manifest `item` is required to have a Media Overlay associated with it. If an EPUB Content Document is wholly or partially referenced by a Media Overlay, then its manifest `item` entry must indicate this via the `media-overlay` attribute.

This is a forwards-compatible addition: 2.0 Reading Systems may safely ignore the `media-overlay` attribute and process documents in their normal fashion.

› 3.5.2 Media Overlays Metadata Vocabulary

The following tables both define a set of properties for use in Package Document metadata and constitute a referenceable vocabulary.

The base IRI for referencing this vocabulary is `http://www.idpf.org/epub/vocab/overlays/#`.

> NOTE
>
> The prefix media: is reserved by [Publications30] for the inclusion of these properties in package metadata.

› active-class	
Description:	Author-defined CSS class name to apply to the currently-playing EPUB Content Document element.
Allowed value(s):	`xsd:string`
Cardinality:	`zero or one`
Example:	`<meta property="media:active-class">-epub-media-overlay-active</meta>`

› duration	
	The duration of the entire presentation or of a specific Media Overlay. The

Description:	specified durations account for the audio clips known at authoring time, and so exclude live streaming from external resources and speech synthesis.
Allowed value(s):	A clock value. Clock values are a subset of SMIL clock values, defined in [SMIL]. See Appendix B, *Examples of Clock Values*.
Cardinality:	Exactly one for the Publication and for each Media Overlay.
Example:	`<meta property="media:duration">1:36:20</meta>`

› narrator

Description:	Name of the narrator.
Allowed value(s):	`xsd:string`
Cardinality:	`zero or more`
Example:	`<meta property="media:narrator">Joe Speaker</meta>`

The Package Document must include the `duration` of each Media Overlay as well as of the entire Publication. The Package Document may include `narrator` information, as well, in particular when each Media Overlay has its own narrator or there is one narrator specified for the entire Publication. The Package Document may also include an author-defined CSS class name to be applied to the currently-playing EPUB Content Document element.

When a `meta` element is specific to a single Media Overlay Document, the `about` attribute is used to reference which one. A `meta` element without an `about` attribute is considered to be about the entire Publication. The `active-class` property must not be used in conjunction with an `about` attribute, as it is always considered to apply to the entire Publication.

The following example shows a Package Document with metadata about Media Overlays.

```
<package>
    <metadata>
        …
        <meta property="media:duration"
refines="#ch1_audio">0:32:29</meta>
        <meta property="media:duration"
refines="#ch2_audio">0:34:02</meta>
        <meta property="media:duration"
refines="#ch3_audio">0:29:49</meta>
        <meta property="media:duration">1:36:20</meta>
        <meta property="media:narrator">Joe Speaker</meta>
        <meta property="media:active-class">-epub-media-overlay-
active</meta>
        …
    </metadata>
    …
</package>
```


› 4 Playback Behaviors

› 4.1 Loading the Media Overlay

When an EPUB Reading System loads a Package Document, it must refer to the manifest `item` elements' `media-overlay` attributes to discover the corresponding Media Overlays for EPUB Content Documents. Playback must start at the Media Overlay element which corresponds to the desired EPUB Content Document starting point. Note that the start of an EPUB Content Document may correspond to an element at the start or in the middle of a Media Overlay. When the Media Overlay Document has finished playing, the Reading System should load the next EPUB Content Document (as specified in the Package Document spine) and also load its corresponding Media Overlay Document, provided that one is given.

› 4.2 Basic Playback

› 4.2.1 Timing and Synchronization

Reading Systems must render immediate children of the `body` element in a sequence. A `seq` element's children must be rendered in sequence, and playback completes when the last child has finished playing. A `par` element's children must be rendered in parallel (with each starting at the same time), and playback completes when all the children have finished playing. When the `body` element's last child has finished playing, playback of the Media Overlay Document is done.

› 4.2.2 Rendering Audio

When presented with a Media Overlay `audio` element, Reading Systems must play the audio resource referenced by the `src` attribute, starting at the clip offset time given by the `clipBegin` attribute and ending at the clip offset time given by the `clipEnd` attribute. The following rules must be observed:

- If `clipBegin` is not specified, its value is assumed to be `0`.
- If `clipEnd` is not specified, its value is assumed to be the full duration of the physical media.
- If `clipEnd` exceeds the full duration of the physical media, then its value is assumed to be the full duration of the physical media.

User-controllable audio playback options should include timescale modification, in which the playback rate is altered without distorting the pitch. The suggested range is half-speed to double-speed.

› 4.2.3 Rendering EPUB Content Document Elements

When presented with a Media Overlay `text` element, Reading Systems should ensure the EPUB Content Document element referenced by the `src` attribute is visible in the Viewport. Reading Systems with a CSS Viewport should add the class name given by the metadata property `active-class` to the actively-playing EPUB Content Document element. Conversely, the class name should be removed when the element is no longer active.

The `active-class` metadata property is optional, and if omitted, Reading System behavior is implementation-specific.

› 4.3 Interacting with the EPUB Content Document

› 4.3.1 Navigation

Because the Media Overlay is closely linked to the EPUB Content Document, it is very easy for Reading Systems to locate a position in the EPUB Content Document based on the current position in the Media

Overlay playback. If the User pauses synchronized playback and navigates to a different part of the Publication, synchronized playback must resume at that point. For example, if a specific page number in the EPUB Content Document is the desired location, then this same point is located in the Media Overlay and playback started there.

This same approach allows for synchronizing the Media Overlay playback with User selection of a navigation points in the EPUB Navigation Document. The Reading System loads the Media Overlay for that file and finds the correct point for starting playback based on the ID [XML] of the navigation point target.

NOTE

A Media Overlay Document may also be associated directly with a Navigation Document in order to provide synchronized playback of its contents, regardless of whether the XHTML Content Document in which it resides is included in the spine. The Reading System should keep playback of the Navigation Document's Media Overlay synchronized with the User's current position in the EPUB Content Document.

› NOTE

Media Overlay Document elements may be associated with EPUB Content Document structures such as tables. Reading Systems should ensure that Media Overlay playback remains synchronized with User navigation of table rows and cells. The Reading System may also play the corresponding table header preceding the contents of the cell.

› 4.3.2 Embedded Audio and Video

An EPUB Content Document with which a Media Overlay is associated may itself contain embedded video and audio media, which may be pointed to by Media Overlay elements. Unlike text and images, video and audio media has an intrinsic duration. Consequently, when a Reading System renders the synchronization described by a Media Overlay, the default playback behavior of audio and video media embedded within the associated EPUB Content Document must be overridden.

Note that the rules below apply only to *referenced* [HTML5] `video` or `audio` elements within the associated EPUB Content Document. That is to say, the rules apply to only those elements pointed to by `text` elements within the Media Overlay (i.e., via the `src` attribute). Embedded media that is not referenced by Media Overlay elements is not subject to these rules.

- All referenced audio and video media embedded within an EPUB Content Document must have their public playback interface deactivated (typically: play/pause control, time slider, volume level, etc.). This behavior is required to avoid interference between the scheduled playback sequence defined by the Media Overlay, and the arbitrary playback behavior due to User interaction or script execution. As a result, when the Reading System is in playback mode, it should:
    - Hide the individual video/audio UI controls from the page, which overrides the default behavior defined by the [HTML5] `controls` attribute.
    - Prevent scripts embedded within the EPUB Content Document from invoking the JavaScript audio/video playback API (i.e., authored as part of the default Publication behavior). It is recommended that content producers should avoid publishing embedded scripts dedicated to controlling the playback of embedded audio/video media. The published Media Overlay can then retain full control of the synchronized presentation without any risk of interference from script-enabled custom behaviors.
- All referenced audio and video media embedded within an EPUB Content Document must be initialized to their "stopped" state, and be ready to be played from the zero-position within their content stream (possibly displaying the image specified using the [HTML5] `poster` attribute). This

requirement overrides the default behavior defined by the [HTML5] `autoplay` attribute.

- When an EPUB Content Document element becomes active, the EPUB Style Sheet visual highlighting rules apply regardless of the content type referred to by that element's `src` attribute (e.g., the CSS class name defined by the active-class metadata property should be applied to visible video and audio player controls within the host EPUB Content Document).

- In addition to the default behavior of Media Overlay activation for textual fragments and images, audio and video playback must be started and stopped according to the duration implied by the authored Media Overlay synchronization (as per the standard [SMIL] timing model). There are two possible scenarios:

    - When a Media Overlay `text` element has no `audio` sibling within its `par` parent container, the referenced EPUB Content Document audio or video media must play until it ends, at which point the `text` element's lifespan terminates. In this case, the implicit duration of the `text` element (and by inference, of the parent `par` container) is that of the referenced audio or video clip.

    - When a Media Overlay `text` element has an `audio` sibling within its `par` parent container, the playback duration of the referenced EPUB Content Document audio or video media must be constrained by the duration of the `audio` sibling. In this case, the actual duration of the parent `par` container is that of the child audio clip, regardless of the duration of the video or audio media pointed to by the `text` element. This behavior may result in embedded video or audio media ending playback prematurely (before reaching its full duration), or ending before the playback of the parallel Media Overlay `audio` is finished (in which case the last-played video frame should remain visible until the parent `par` container finally ends). This behavior is equivalent of the Media Overlay `audio` element implicitly carrying the behavior of the [SMIL] `endsync` attribute.

      Furthermore, Reading Systems should expose User controls for the volume levels of each independent audio track (i.e., from the `audio` element of the Media Overlay, and from the embedded audio or video media within the EPUB Content Document), so that audio output can be adjusted to match listeners' requirements. Note that having overlapping audio tracks is typically an authoring-time concern: content producers usually add a layer of audio information over a video track for description purposes. It is recommended that overlapping audio situations are carefully examined and dealt with at production stage, as Reading Systems are not required to handle simultaneous volume levels in any particular way.

- When a `text` element becomes inactive in the Media Overlay, and when it points to embedded video or audio media, that referenced media must be reset to its initial "stopped" state, ready to be played from the zero-position within their content stream (possibly displaying the poster image specified using the HTML5 markup).

› 4.3.3 Text-to-Speech

When a Media Overlay `text` element with no `audio` sibling element references text within the target EPUB Content Document, Reading Systems capable of Text-to-Speech (TTS) should render the referenced text using TTS.

As per Reading System conformance requirements, the speech-related information provided in the target EPUB Content Document should be used to play the audio stream as part of the Media Overlay rendering. See Reading System Text-to-Speech Conformance Requirements [Publications30].

The Media Overlay `text` element's lifespan corresponds to the rendering time of the associated speech synthesis. The implicit duration of the `text` element (and by inference, of the parent `par` element) is therefore determined by the execution of the Text-to-Speech engine, and cannot be known at authoring time (factors like speech rate, pauses and other prosody parameters influence the audio output).

› 4.4 Skippability and Escapability

› 4.4.1 Skippability

While reading, Users may want to turn on or off certain features of the Publication, such as sidebars, footnotes, page numbers, or other types of secondary content. This feature is called skippability. Reading Systems should use the semantic information provided by Media Overlay elements' epub:type attribute to determine when to offer Users the option of skippable features. In the following example, a Reading System should offer the User the option of turning on and off the page break/page number announcements, which are often cumbersome to listen to.

The following example shows a Media Overlay Document with a pagebreak.

```
<smil xmlns="http://www.w3.org/ns/SMIL"
      xmlns:epub="http://www.idpf.org/2007/ops"
      version="3.0">
    <body>
        <!-- a paragraph -->
        <par id="id1">
            <text src="chapter1.xhtml#para1"/>
            <audio src="chapter1_audio.mp3" clipBegin="0:23:22.000"
clipEnd="0:24:15.000"/>
        </par>

        <!-- a page number -->
        <par id="id2" epub:type="pagebreak">
            <text src="chapter1.xhtml#pgbreak1"/>
            <audio src="chapter1_audio.mp3" clipBegin="0:24:15.000"
clipEnd="0:24:18.123"/>
        </par>

        <!-- another paragraph -->
        <par id="id3">
            <text src="chapter1.xhtml#para2"/>
            <audio src="chapter1_audio.mp3" clipBegin="0:24:18.123"
clipEnd="0:25:28.530"/>
        </par>
    </body>
</smil>
```

The following example shows an EPUB Content Document with a pagebreak.

```
<html … >
    …
    <body>
        <p id="para1">This is the paragraph before the pagebreak … </p>

        <br id="pgbreak1" epub:type="pagebreak" title="234"/>

        <p id="para2">This is the paragraph after the pagebreak …</p>
    </body>
</html>
```

The following selection of terms from the [StructureVocab] for which Reading Systems should offer Users the option of skippability is provided as an informative reference:

- sidebar
- practice
- marginalia
- annotation
- help
- note
- footnote
- rearnote
- pagebreak

Media Overlays may use additional vocabularies by defining them in the `epub:prefix` attribute on the root `smil` element. Reading System support for skippability based on `epub:type` values should not be assumed.

› 4.4.2 Escapability

Escapable items are nested structures such as tables, lists, and sidebars that listeners may wish to skip over, continuing to read from the point immediately after the nested structure. The escapability feature differs from the skippability feature in that it does not enable or disable entire types of items, but provides an exit from them (e.g., a User can listen to some of the content before choosing to escape). Reading Systems should allow escaping of nested structures. Reading Systems must determine the start of nested structures by the value of the `epub:type` attribute (e.g., `glossary`) and should offer Users the option to skip playback of that structure and resume with whatever content comes after it.

The following example shows the Media Overlay Document for an EPUB Content Document containing a paragraph, a glossary, and another paragraph. A Reading System that supported escapability would give the User the option to interrupt playback of the glossary and continue playing the document paragraphs.

```
<smil xmlns="http://www.w3.org/ns/SMIL"
      xmlns:epub="http://www.idpf.org/2007/ops"
      version="3.0">
    <body>
        <!-- a paragraph, part of the regular document text -->
        <par id="id1">
            <text src="chapter1.xhtml#para1"/>
            <audio src="chapter1_audio.mp3" clipBegin="0:23:22.000"
clipEnd="0:24:15.000"/>
        </par>

        <!-- a glossary, which is a nested structure -->
        <seq id="id2" epub:textref="chapter1.xhtml#g0"
epub:type="glossary">
            <par id="id3" epub:type="glossterm">
                <text src="chapter1.xhtml#g1"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:24:15.000"
clipEnd="0:24:18.123"/>
            </par>
            <par id="id4" epub:type="glossdef">
                <text src="chapter1.xhtml#g2"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:24:18.123"
clipEnd="0:25:28.530"/>
            </par>
            <par id="id5" epub:type="glossterm">
```

```
                <text src="chapter1.xhtml#g3"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:25:28.530"
clipEnd="0:25:45.515"/>
            </par>
            <par id="id6" epub:type="glossdef">
                <text src="chapter1.xhtml#g4"/>
                <audio src="chapter1_audio.mp3" clipBegin="0:25:45.515"
clipEnd="0:27:04.123"/>
            </par>
        </seq>

        <!-- another paragraph, part of the document text that comes
after the glossary -->
        <par id="id7">
            <text src="chapter1.xhtml#para2"/>
            <audio src="chapter1_audio.mp3" clipBegin="0:27:04.123"
clipEnd="0:27:59.000"/>
        </par>
    </body>
</smil>
```

› Appendix A. Media Overlays Schema

The schema for Media Overlays is available at ../schema/media-overlay-30.nvdl.

This schema is normative.

› A.1 Using the Media Overlays Schema

This section is informative

Validation of Media Overlays using this schema will require a processor that supports [NVDL], [RelaxNG] and [ISOSchematron].

Note, however, that the NVDL schema layer can be substituted by a two-pass validation using the embedded RELAX NG and ISO Schematron schemas alone.

› Appendix B. Examples of Clock Values

This appendix is informative

The following are examples of allowed clock values:

- `5:34:31.396` = 5 hours, 34 minutes, 31 seconds and 396 milliseconds
- `124:59:36` = 124 hours, 59 minutes and 36 seconds
- `0:05:01.2` = 5 minutes, 1 second and 200 milliseconds
- `0:00:04` = 4 minutes