Multimedia domain

Targeted DITAweb review

July 2019

Table of contents

1 Review information

Here are some general guidelines:

1. Review the content as distributed in PDF format, and then enter your comments in DITAweb at . DITAweb can introduce some formatting inconsistencies, so ensure that any formatting errors – such as extra spaces – are present in the PDF before commenting in DITAweb.
2. For early targeted reviews, the attributes topics are not included in the review, although cross references to them are rendered in the PDF output. If the attributes topics are available in DITAweb, do not comment on them.
3. The DITA Technical Committee uses the IBM Style Guide.
4. Contact Alan Houser if you need a password to be reset or a new account created in DITAweb.

2 Multimedia domain elements

The multimedia elements are used to reference audio or video content. The elements in this domain are modeled on the HTML5 `<audio>` and `<video>` elements.

2.1 <audio>

Audio objects reference sound content.

Usage information

Audio objects are modeled on the HTML5 `<audio>` element.

Audio objects can be referenced by `@data`, `@datakeyref`, and nested `<media-source>` elements. Nested `<media-source>` elements enable extensive configuration of how the audio object is presented.

Rendering expectations

When an audio object cannot be rendered in a meaningful way, processors **SHOULD** present the contents of the `<fallback>` element, if it is present.

Processing expectations

Behaviours such as auto-playing, looping, and muting are determined by child elements. When not specified, the behavior depends on the user agent.

Specialization hierarchy

The `<audio>` element is specialized from `<object>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@data

Specifies the absolute or relative URI of the audio object. If this attribute is specified, `@type` also should be specified.

@datakeyref

Provides a key reference to the audio object. When specified and the key is resolvable, the key-provided URI is used. If the key referenced by `@datakeyref` cannot be resolved, and `@data` is specified, that value provided by `@data` is used as a fallback. If the key referenced by `@datakeyref` has no associated resource, only link text, and the audio object does not contain a `<fallback>` element, the link text becomes fallback content for the audio object.

Comment by Kristen J Eberlein on 17 May 2019

Robert and I discussed and reworked the above description today, in response to feedback from Chris Nitchie.

However, this content is really about key resolution; we need to ensure that this is covered clearly (and probably with normative language) in the primary topics about key resolution.

@tabindex

Positions the audio object in tabbing order.

Comment by Kristen J Eberlein on 28 April 2019

Would this be a better description? "Indicates whether the audio object can be focused and where it participates in sequential keyboard navigation."

@type

Indicates the MIME type for the audio object. This attribute enables processors to avoid loading unsuppported objects. If `@type` is not specified, the effective type for the key named by the `@datakeyref` attribute is used as the value. If an explicit `@type` is not specified on either the audio object or key definition, processors can use other means, such the URI file extension, to determine the effective MIME type of the media object.

Examples

In the following code sample, an audio object is referenced using direct addressing. The `@type` attribute specifies the MIME type of the object.

```
<audio data="message.mp3" type="audio/mp3"/>
```

The audio object also could be addressed using a key reference; in this version, both the URI and the MIME type come from the key definition:

```
<audio datakeyref="message"/>
```

Figure 1: A simple audio object

In the following code sample, `<media-source>` elements are used to specify the different audio formats that are available.

```
<audio>
  <media-source value="message.mp3" type="audio/mp3"/>
  <media-source value="message.wav" type="audio/wav"/>
</audio>
```

Figure 2: An audio object with multiple formats

The following code sample specifies an audio object and defines multiple presentational details; it also provides fallback behavior for when the audio cannot be rendered.

```
<audio>
  <desc>A sound file narrating the execution of this procedure.</desc>
  <fallback>The audio track walking through this procedure is not available.</fallback>

  <!--
    When the following elements are used, they have a default value of "true";
    setting value="true" and not specifying @value have the same effect.
    To disable any of these settings, specify value="false". -->

  <media-controls value="true"/>
  <media-autoplay/>
  <media-loop value="false"/>
  <media-muted value="false"/>
```

```
  <!-- Multiple formats, with URI and MIME type referenced using a key -->
  <media-source keyref="walkthrough-mp3"/>
  <media-source keyref="walkthrough-wav"/>

</audio>
```

Figure 3: Complex example of an audio object

2.2 <media-autoplay>

Autoplay settings control whether referenced media plays automatically.

Usage information

Autoplay settings are modeled on the `@autoplay` attribute on HTML5 media elements. If autoplay settings are not present, the default behavior is determined by the user agent that is used to present the media.

Specialization hierarchy

The `<media-autoplay>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@name
The value is fixed to "autoplay".

@value
Specifies whether the media object automatically plays when the topic is displayed. The following values are recognized:

> **Comment by Kristen J Eberlein on 28 April 2019**
>
> Should we replace "when the topic is displayed" with "when the audio media object is rendered"?

true
Default. Auto-playing is enabled.

false
Autoplay is disabled.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Example

See audio (4) and video (10).

2.3 <media-controls>

Media control settings specify whether user-interface components are presented to control the playback of the referenced media

Usage information

Media control settings are modeled on the `@controls` attribute on HTML5 media elements. If media control settings are not present, the default behavior is determined by the user agent that is used to present the media.

Specialization hierarchy

The `<media-controls>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@name
The value is fixed to "controls".

@value
Specifies whether the presentation of the media object includes user interface controls. The following values are recognized:

true
Default. Controls are presented.

false
Controls are not presented.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Example

See audio (4) and video (10).

2.4 <media-loop>

Media loop settings control whether the referenced media restarts automatically when it has completed playing.

Usage information

Media loop settings are modeled on the `@loop` attribute on HTML5 media elements. If media loop settings are not present, the default behavior is determined by the user agent that is used to present the media.

Specialization hierarchy

The `<media-loop>` element is specialized from `<object>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@name
The value is fixed to "loop".

@value
Specifies whether the media object loops when played. The following values are recognized:

true
Default. Looped playback is enabled.

false
Looped playback is disabled

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Example

See audio (4) and video (10).

2.5 <media-muted>

Media mute settings control whether the referenced media plays with or without sound.

Usage information

Media mute settings are modeled on the `@muted` attribute on HTML5 media elements. If media mute settings are not present, the default behavior is determined by the user agent that is used to present the media.

Specialization hierarchy

The `<media-muted>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@name
The value is fixed to "muted".

@value
Specifies whether the media object is muted. The following values are recognized:

true
Default. Playback is muted.

false
Playback is not muted.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

Example

See audio (4) and video (10).

2.6 <media-source>

The media source specifies the location of a representation of the referenced media.

Usage information

The media source is modeled on the `<source>` element used in HTML5 media elements.

Specialization hierarchy

The `<media-source>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@keyref
: A key reference to the media resource. If both `@keyref` and `@value` are specified, the resource referenced by `@keyref` takes precedence unless the key cannot be resolved, in which case the resource specified by `@value` is used as a fallback.

@name
: The value is fixed to "source".

@type
: Specifies the MIME type of the resource specified by `@value`.

@value
: Specifies the URL of the media resource.

@valuetype
: The value is fixed to "ref".

Example

See audio (4) and video (10).

2.7 <media-track>

Media track settings specify the location of supplemental text-based data for the referenced media, for example, subtitles or descriptions.

Usage information

The media track settings are modeled on the `<track>` element used in HTML5 media elements. They refer to track resources that use Web Video Text Track Format (WebVTT).

Specialization hierarchy

The `<media-track>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@keyref
A key reference to the track resource. If both `@keyref` and `@value` are specified, the resource referenced by `@keyref` takes precedence unless the key cannot be resolved, in which case the resource specified by `@value` is used as a fallback.

@name
The value is fixed to "track".

@type
Specifies the usage for the track resource. This attribute is modeled on the `@kind` attribute on the HTML5 `<track>` element, as described by the *W3C HTML5 specification*. The values for this attribute are derived from the HTML5 standard:

captions
Transcription or translation of the dialogue, sound effects, relevant musical cues, and other relevant audio information. This is intended for use when the soundtrack is unavailable (for example, because it is muted or because the user is hard-of-hearing). This information is rendered over the video and labeled as appropriate for hard-of-hearing users.

chapters
Chapter titles, which are intended to be used for navigating the media resource. The chapter titles are rendered as an interactive list in the interface for the user agent.

descriptions
Textual descriptions of the video component of the media resource. This is intended for audio synthesis when the visual component is unavailable (for example, because the user is interacting with the application without a screen or because the user is blind). Descriptions are synthesized as separate audio tracks.

metadata
Tracks intended for use from script. This metadata is not displayed by the user agent.

subtitles
Transcription or translation of the dialogue, suitable for when the sound is available but not understood (for example, because the user does not understand the language of the soundtrack). Subtitles are rendered over the video.

-dita-use-conref-target
See Using the -dita-use-conref-target value for more information.

@value
Specifies the URI of the track resource.

Example

See audio (4) and video (10).

2.8 <video>

Video objects reference moving visual media.

Usage information

Video objects are modeled on the HTML5 `<video>` element.

Video objects can be referenced by `@data`, `@datakeyref`, and nested `<media-source>` elements. Nested `<media-source>` elements enable extensive configuration of how the video object is presented.

Rendering expectations

The video object typically is rendered in the main flow of the content.

Processors **SHOULD** scale the object when values are provided for the `@height` and `@width` attributes. The following expectations apply:

- If a height value is specified and no width value is specified, processors **SHOULD** scale the width by the same factor as the height.
- If a width value is specified and no height value is specified, processors **SHOULD** scale the height by the same factor as the width.
- If both a height value and width value are specified, implementations **MAY** ignore one of the two values when they are unable to scale to each direction using different factors.

When a video object cannot be rendered in a meaningful way, processors **SHOULD** present the contents of the `<fallback>` element, if it is present.

Specialization hierarchy

The `<video>` element is specialized from `<object>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@data
: Specifies the absolute or relative URI of the video object. If this attribute is specified, `@type` also should be specified.

@datakeyref
: Provides a key reference to the video object. When specified and the key is resolvable, the key-provided URI is used. If the key referenced by `@datakeyref` cannot be resolved, and `@data` is specified, that value provided by `@data` is used as a fallback. If the key referenced by `@datakeyref` has no associated resource, only link text, and the video object does not contain a `<fallback>` element, the link text becomes fallback content for the video object.

@height
: Indicates the vertical dimension for the resulting display. The value of this attribute is a real number (expressed in decimal notation) optionally followed by a unit of measure from the set of pc, pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels). Possible values include: "5", "5in", and "10.5cm".

@type
: Indicates the MIME type for the video object. This attribute enables processors to avoid loading unsuppported objects. If `@type` is not specified, the effective type for the key named by the `@datakeyref` attribute is used as the value. If an explicit `@type` is not specified on either the audio object or key definition, processors can use other means, such the URI file extension, to determine the effective MIME type of the media object.

@width
: Indicates the horizontal dimension for the resulting display. The value of this attribute is a real number (expressed in decimal notation) optionally followed by a unit of measure from the set of pc,

pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels). Possible values include: "5", "5in", and "10.5cm".

@tabindex
Position the video in tabbing order.

Examples

In the following code sample, a video object is referenced using direct addressing. The `@type` attribute specifies the MIME type of the object.

```
<video data="video.mp4" type="video/mp4"/>
```

The video object also could be addressed using a key reference; in this version, both the URI and the MIME type come from the key definition.

```
<video datakeyref="video"/>
```

Figure 4: A simple video object

In the following code sample, `<media-source>` elements are used to specify the different video formats that are available.

```
<video>
  <media-source value="video.mp4" type="video/mp4"/>
  <media-source value="video.ogg" type="video/ogg"/>
  <media-source value="video.webm" type="video/webm"/>
</video>
```

Figure 5: A video object with multiple formats

The following code sample defines multiple presentational details for a video that is available in multiple formats. The video is referenced using key reference and a fallback image is provided for use when the video cannot be displayed.

```
<video width="400px" height="300px">
  <desc>A video illustrating this procedure.</desc>
  <fallback>
    <image href="video-not-available.png">
      <alt>This video cannot be displayed.</alt>
    </image>
  </fallback>

  <!-- Reference the poster using a key -->
  <video-poster keyref="video-poster"/>

  <!--
    When the following elements are used, they have a default value of "true";
    setting value="true" and not specifying @value have the same effect.
    To turn any of these settings off, specify value="false".
  -->
  <media-controls value="true"/>
  <media-autoplay/>
  <media-loop value="false"/>
  <media-muted value="false"/>

  <!-- Multiple formats, referenced via key. The key definition
       specifies both the URI and the MIME type -->
  <media-source keyref="video-mp4"/>
  <media-source keyref="video-ogg"/>
  <media-source keyref="video-webm"/>

  <!-- Subtitle tracks in English, French and German.
       Each key definition provides a URI and specifies the type value "subtitles". -->
  <media-track xml:lang="en" keyref="video-subtitles-en"/>
  <media-track xml:lang="fr" keyref="video-subtitles-fr"/>
```

```
   <media-track xml:lang="de" keyref="video-subtitles-de"/>
 </video>
```

Figure 6: Complex example of a video object, with multiple formats and multi-lingual subtitles

2.9 <video-poster>

Video poster settings control the image that is rendered before video playback begins.

Usage information

The video poster settings are modeled on the `@poster` attribute on the HTML5 `<video>` element. If video poster settings are not present, the default behavior is determined by the user agent that is used to present the media.

Specialization hierarchy

The `<video-poster>` element is specialized from `<param>`. It is defined in the multimedia-domain module.

Attributes

The following attributes are available on this element: Universal attribute group and the attributes defined below.

@keyref
: Provides a key reference to the image. If both `@keyref` and `@value` are specified, the image referenced by `@keyref` takes precedence unless the key cannot be resolved, in which case the resource specified by `@value` is used as a fallback.

@name
: The value is fixed to "poster".

@type
: Specifies the MIME type of the resource specified by `@value`.

@value
: Specifies the URL of the image.

@valuetype
: The value is fixed to "ref".

Example

See `<video>` (10) for examples of this element.