



DITA Technical Committee

Stage three: Issue #351 Add multimedia elements to base

Table of contents

1 Stage three: #351 Add multimedia elements to base.....	3
<audio>.....	12
<media-source>.....	14
<media-track>.....	14
<video>.....	15
Index.....	19

1 Stage three: #351 Add multimedia elements to base

Add multimedia elements to base.

Champion

Kristen James Eberlein, Eberlein Consulting LLC

Tracking information

Event	Date	Links
Stage 1 proposal accepted	24 March 2020	Minutes, 24 March 2020
Stage 2 proposal submitted	29 June 2020 Revision #1: 28 July 2020	Original Revision #1
Stage 2 proposal discussed	07 July 2020 28 July 2020	Minutes, 07 July 2020 Minutes, 28 July 2020
Stage 2 proposal approved	18 August 2020	Minutes, 18 August 2020
Stage 3 proposal submitted to reviewers	22 August 2020	Carlos Evia, Virginia Tech Chris Nitchie, Individual member
Stage 3 proposal (this document) submitted to TC	01 September 2020	

Approved technical requirements

Four new, unspecialized elements will be added to the DITA 2.0 base:

<audio>

Description: Audio is sound that the human ear is capable of hearing.

Content model:

- Zero or one <desc>, <longdescref>, and <fallback>
- Zero or more <media-source>, <media-track>, and %foreign.unknown.incl;

Contained by: Everywhere that <object> is available

Attributes:

- Universal attributes
- @href, @keyref, and @type
- @autoplay, @controls, @loop, @muted, and @tabindex

<media-source>

Description: The media source specifies the location of an audio or video resource.

Content model: Empty

Contained by: <audio> and <video>

Attributes:

- Universal attributes
- @href, @keyref, and @type

<media-track>

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

Content model: PCDATA (to specify a user-readable title of the track text, to be used by browsers when listing available text tracks). This corresponds to the HTML5 attribute @label.

Contained by: <audio> and <video>

Attributes:

- Universal attributes
- @srclang: Specifies the language of the track text
- @href, @keyref, and @kind

<video>

Description: A video is a recording of moving visual images.

Content model:

- Zero or one <desc> and <fallback>
- Zero or more <media-source>
- Zero or more <media-track>

Contained by: Everywhere that <object> is available

Attributes:

- Universal attributes
- @href, @keyref, and @type
- @height and @width
- @autoplay, @controls, @loop, @muted, @poster, and @tabindex
- @posterkeyref

Dependencies or interrelated proposals

None

Removed grammar files

The following files must be removed:

- dtd/base/mediaDomain.ent
- dtd/base/mediaDomain.mod
- rng/base/mediaDomain.rng

The following files must be modified to remove references to the removed files:

- dtd/base/catalog.xml
- rng/base/catalog.xml

Modified grammar files

The following files must be modified:

- (DITD) commonElements.ent
- (DITD) commonElements.mod
- (RNG) commonElements.rng

In the content below, the following conventions are used:

- Bold is used to indicate code to be added, for example, **addition**.
- Line-through and red text is used to indicate code to be removed, for example, ~~removal~~.
- Ellipses (...) indicate where code is snipped for brevity.

Figure 1: Changes to commonElements.ent

```

...
<!-- ===== -->
<!-- ELEMENT NAME ENTITIES -->
<!-- ===== -->
...
<!ENTITY % audio          "audio"          >
<!ENTITY % video          "video"          >
<!ENTITY % media-source   "media-source"   >
<!ENTITY % media-track    "media-track"    >
...

```

Figure 2: Changes to commonElements.mod

```

...
<!-- ===== -->
<!-- ELEMENT DECLARATIONS -->
<!-- ===== -->
...
<!-- LONG NAME: Audio -->
<!ENTITY % audio.content
    "( (%desc;)?,
      (%longdescref;)?,
      (%fallback;)?,
      (%media-source;)*,
      (%media-track;)*,
      (%foreign.unknown.incl;)* )"
>
<!ENTITY % audio.attributes
    "autoplay
      (true |
        false |
        -dita-use-conref-target)
        #IMPLIED

    controls
      (true |
        false |
        -dita-use-conref-target)
        #IMPLIED

    loop
      (true |
        false |
        -dita-use-conref-target)
        #IMPLIED

    muted
      (true |
        false |
        -dita-use-conref-target)
        #IMPLIED

    href
      CDATA
        #IMPLIED

    keyref
      CDATA
        #IMPLIED

    type
      CDATA
        #IMPLIED

    tabindex
      NMTOKEN
        #IMPLIED

    %univ-atts;"
>
<!ELEMENT audio %audio.content;>

```

```

<!ATTLIST audio %audio.attributes;>

<!-- LONG NAME: Video -->
<!ENTITY % video.content
      "( (%desc;)?,
         (%longdescref;)?,
         (%fallback;)?,
         (%media-source;)*,
         (%media-track;)*,
         (%foreign.unknown.incl;)* )"
>
<!ENTITY % video.attributes
      "autoplay
        (true |
         false |
         -dita-use-conref-target)
         #IMPLIED

        controls
        (true |
         false |
         -dita-use-conref-target)
         #IMPLIED

        loop
        (true |
         false |
         -dita-use-conref-target)
         #IMPLIED

        muted
        (true |
         false |
         -dita-use-conref-target)
         #IMPLIED

        poster
        CDATA
         #IMPLIED

        posterkeyref
        CDATA
         #IMPLIED

        href
        CDATA
         #IMPLIED

        keyref
        CDATA
         #IMPLIED

        type
        CDATA
         #IMPLIED

        tabindex
        NMTOKEN
         #IMPLIED

        %univ-atts;"
>
<!ELEMENT video %video.content;>
<!ATTLIST video %video.attributes;>

<!-- LONG NAME: Media source -->
<!ENTITY % media-source.content
      "EMPTY"
>
<!ENTITY % media-source.attributes
      "href
        CDATA
         #IMPLIED

        keyref
        CDATA
         #IMPLIED

        type
        CDATA
         #IMPLIED

        %univ-atts;"
>
<!ELEMENT media-source %media-source.content;>

```

```

<!ATTLIST media-source %media-source.attributes;>

<!-- LONG NAME: Media track -->
<!ENTITY % media-track.content
        "(#PCDATA)"
>
<!ENTITY % media-track.attributes
        "href
                CDATA
                #IMPLIED
        keyref
                CDATA
                #IMPLIED
        kind
                (subtitles |
                 captions |
                 descriptions |
                 chapters |
                 metadata |
                 -dita-use-conref-target)
                #IMPLIED
        srclang
                CDATA
                #IMPLIED
        %univ-atts;"
>
<!ELEMENT media-track %media-track.content;>
<!ATTLIST media-track %media-track.attributes;>
...
<!-- ===== -->
<!-- SPECIALIZATION ATTRIBUTE DECLARATIONS -->
<!-- ===== -->
...
<!ATTLIST audio class CDATA "- topic/audio ">
<!ATTLIST video class CDATA "- topic/video ">
<!ATTLIST media-source class CDATA "- topic/media-source ">
<!ATTLIST media-track class CDATA "- topic/media-track ">
...

```

In addition, <audio> and <video> must be added to the following entities:

- %basic.block;
- %basic.blocknotbl;
- %basic.block.nonote;
- %basic.block.nopara;
- %basic.block.nolq;
- %basic.block.notbnofg;
- %data.cnt;

Figure 3: Changes to commonElementsMod.rng

```

...
<div>
  <a:documentation>ELEMENT TYPE NAME PATTERNS</a:documentation>
  ...
  <define name="audio">
    <ref name="audio.element"/>
  </define>
  <define name="video">
    <ref name="video.element"/>
  </define>
  <define name="media-source">
    <ref name="media-source.element"/>
  </define>
  <define name="media-track">
    <ref name="media-track.element"/>
  </define>
  ...
</div>

```

```

<div>
...
<div>
<a:documentation xml:space="preserve"> LONG NAME: Audio</a:documentation>
<define name="audio.content">
  <optional>
    <ref name="desc"/>
  </optional>
  <optional>
    <ref name="longdescref"/>
  </optional>
  <optional>
    <ref name="fallback"/>
  </optional>
  <zeroOrMore>
    <ref name="media-source"/>
  </zeroOrMore>
  <zeroOrMore>
    <ref name="media-track"/>
  </zeroOrMore>
  <zeroOrMore>
    <ref name="foreign.unknown.incl"/>
  </zeroOrMore>
</define>
<define name="audio.attributes">
  <optional>
    <attribute name="autoplay">
      <choice>
        <value>true</value>
        <value>false</value>
        <value>-dita-use-conref-target</value>
      </choice>
    </attribute>
  </optional>
  <optional>
    <attribute name="controls">
      <choice>
        <value>true</value>
        <value>false</value>
        <value>-dita-use-conref-target</value>
      </choice>
    </attribute>
  </optional>
  <optional>
    <attribute name="loop">
      <choice>
        <value>true</value>
        <value>false</value>
        <value>-dita-use-conref-target</value>
      </choice>
    </attribute>
  </optional>
  <optional>
    <attribute name="muted">
      <choice>
        <value>true</value>
        <value>false</value>
        <value>-dita-use-conref-target</value>
      </choice>
    </attribute>
  </optional>
  <optional>
    <attribute name="href"/>
  </optional>
  <optional>
    <attribute name="keyref"/>
  </optional>
  <optional>
    <attribute name="type"/>
  </optional>
  <optional>
    <attribute name="tabindex">
      <data type="NMTOKEN"/>
    </attribute>
  </optional>
</define>

```



```

</optional>
<ref name="univ-attns"/>
</define>
<define name="audio.element">
  <element name="audio" dita:longName="Audio">
    <a:documentation>DITA's <code><audio></code> element corresponds to the HTML <code><audio></code> element.
      Category: Body elements</a:documentation>
    <ref name="audio.attlist"/>
    <ref name="audio.content"/>
  </element>
</define>
<define name="audio.attlist" combine="interleave">
  <ref name="audio.attributes"/>
</define>
</div>
<div>
  <a:documentation xml:space="preserve"> LONG NAME: Video</a:documentation>
  <define name="video.content">
    <optional>
      <ref name="desc"/>
    </optional>
    <optional>
      <ref name="longdescref"/>
    </optional>
    <optional>
      <ref name="fallback"/>
    </optional>
    <zeroOrMore>
      <ref name="media-source"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="media-track"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="foreign.unknown.incl"/>
    </zeroOrMore>
  </define>
  <define name="video.attributes">
    <optional>
      <attribute name="autoplay">
        <choice>
          <value>true</value>
          <value>false</value>
          <value>-dita-use-conref-target</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="controls">
        <choice>
          <value>true</value>
          <value>false</value>
          <value>-dita-use-conref-target</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="loop">
        <choice>
          <value>true</value>
          <value>false</value>
          <value>-dita-use-conref-target</value>
        </choice>
      </attribute>
    </optional>
    <optional>
      <attribute name="muted">
        <choice>
          <value>true</value>
          <value>false</value>
          <value>-dita-use-conref-target</value>
        </choice>
      </attribute>
    </optional>
  </define>

```

```

<optional>
  <attribute name="poster"/>
</optional>
<optional>
  <attribute name="href"/>
</optional>
<optional>
  <attribute name="keyref"/>
</optional>
<optional>
  <attribute name="posterkeyref"/>
</optional>
<optional>
  <attribute name="type"/>
</optional>
<optional>
  <attribute name="height">
    <data type="NMTOKEN"/>
  </attribute>
</optional>
<optional>
  <attribute name="width">
    <data type="NMTOKEN"/>
  </attribute>
</optional>
<optional>
  <attribute name="tabindex">
    <data type="NMTOKEN"/>
  </attribute>
</optional>
<ref name="univ-atts"/>
</define>
<define name="video.element">
  <element name="video" dita:longName="Video">
    <a:documentation>DITA's <video> element corresponds to the HTML <video> element.
      Category: Body elements</a:documentation>
    <ref name="video.attlist"/>
    <ref name="video.content"/>
  </element>
</define>
<define name="video.attlist" combine="interleave">
  <ref name="video.attributes"/>
</define>
</div>

<div>
  <a:documentation xml:space="preserve"> LONG NAME: Media source </a:documentation>
  <define name="media-source.content">
    <empty/>
  </define>
  <define name="media-source.attributes">
    <optional>
      <attribute name="href"/>
    </optional>
    <optional>
      <attribute name="keyref"/>
    </optional>
    <attribute name="type"/>
    <ref name="univ-atts"/>
  </define>
  <define name="media-source.element">
    <element name="media-source" dita:longName="Audio">
      <a:documentation>DITA's <media-source> element corresponds to the HTML <source>
        element within <video> and <audio> elements.</a:documentation>
      <ref name="media-source.attlist"/>
      <ref name="media-source.content"/>
    </element>
  </define>
  <define name="media-source.attlist" combine="interleave">
    <ref name="media-source.attributes"/>
  </define>
</div>

<div>

```

```

<a:documentation xml:space="preserve"> LONG NAME: Media track </a:documentation>
<define name="media-track.content">
  <text/>
</define>
<define name="media-track.attributes">
  <optional>
    <attribute name="href"/>
  </optional>
  <optional>
    <attribute name="keyref"/>
  </optional>
  <optional>
    <attribute name="kind">
      <choice>
        <value>subtitles</value>
        <value>captions</value>
        <value>descriptions</value>
        <value>chapters</value>
        <value>metadata</value>
        <value>-dita-use-conref-target</value>
      </choice>
    </attribute>
  </optional>
  <optional>
    <attribute name="srclang"/>
  </optional>
  <ref name="univ-atts"/>
</define>
<define name="media-track.element">
  <element name="media-track" dita:longName="Audio object reference">
    <a:documentation>DITA's <code><media-track></code> element corresponds to the HTML <code><track></code>
    element within <code><video></code> and <code><audio></code> elements.</a:documentation>
    <ref name="media-track.attlist"/>
    <ref name="media-track.content"/>
  </element>
</define>
<define name="media-track.attlist" combine="interleave">
  <ref name="media-track.attributes"/>
</define>
...
</div>
<div>
  <a:documentation> Specialization attributes. Global attributes and class defaults </
a:documentation>
  ...
  <define name="audio.attlist" combine="interleave">
    <optional>
      <attribute name="class"
        a:defaultValue="- topic/audio "/>
    </optional>
  </define>
  <define name="video.attlist" combine="interleave">
    <optional>
      <attribute name="class"
        a:defaultValue="- topic/video "/>
    </optional>
  </define>
  <define name="media-source.attlist" combine="interleave">
    <optional>
      <attribute name="class"
        a:defaultValue="- topic/media-source "/>
    </optional>
  </define>
  <define name="media-track.attlist" combine="interleave">
    <optional>
      <attribute name="class"
        a:defaultValue="- topic/media-track "/>
    </optional>
  </define>
  ...
</div>
</grammar>

```

In addition, `<audio>` and `<video>` must be added to the following patterns:

- `basic.block`
- `basic.blocknotbl`
- `basic.block.nonote`
- `basic.block.nopara`
- `basic.block.nolq`
- `basic.block.notbnofg`
- `data.cnt`

Modified terminology

None

Modified specification documentation

The following element-reference topics will be added to the specification:

- [audio](#) (12)
- [media-source](#) (14)
- [media-track](#) (14)
- [video](#) (15)

The current topics for the multimedia domain (and any references to them) will be removed.

The following topic will need to be modified: "Element-by-element recommendations for translators."

Migration plans for backwards incompatibilities

Not applicable.

`<audio>`

Audio is sound that the human ear is capable of hearing.

Usage information

The `<audio>` element is modeled on the HTML5 `<audio>` element.

An audio resource can be referenced by `@href`, `@keyref`, and nested `<media-source>` elements.

Behaviors such as auto-playing, looping, and muting are determined by attributes. When not specified, the default behavior is determined by the user agent that is used to present the media.

Rendering expectations

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

Attributes

The following attributes are available on this element: [Universal attribute group](#) and the attributes defined below.

@autoplay

Specifies whether the resource automatically plays when it is presented. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@controls

Specifies whether the presentation of the resource includes user interface controls. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@href

Specifies the absolute or relative URI of the audio resource. If @href is specified, specify @type also.

@keyref

Specifies a key reference to the audio resource.

@loop

Specifies whether the resource loops when played. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@muted

Specifies whether the resource is muted. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@tabindex

Specifies whether the audio resource can be focused and where it participates in sequential keyboard navigation. See [@tabindex](#) in the HTML specification (WHATWG version).

@type

Specifies the MIME type for the audio resource. This attribute enables processors to avoid loading unsupported resources. If @type is not specified and @keyref is specified, the effective type for the key named by the @keyref attribute is used as the value. If an explicit @type is not specified on either the <audio> element or key definition, processors can use other means, such the URI file extension, to determine the effective MIME type of the audio resource.

Examples

Figure 4: An <audio> element that uses direct addressing

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

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

Figure 5: An <audio> element that uses indirect addressing

In the following code sample, the audio resource is addressed using a key reference:

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

Both the URI and the MIME type are specified on the key definition:

```
<keydef keys="message" href="message.mp3" type="audio/mp3"/>
```

Figure 6: An <audio> element with multiple formats

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

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

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

Figure 7: Example of a complex <audio> element

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

```
<audio autoplay="true"
        controls="true"
        loop="false"
        muted="false">
  <desc>A sound file narrating the performance of this procedure.</desc>
  <fallback>The audio track walking through this procedure is not available.</fallback>
  <!-- Multiple formats, with URI and MIME type referenced using a key -->
  <media-source keyref="walkthrough-mp3"/>
  <media-source keyref="walkthrough-wav"/>
</audio>
```

<media-source>

The media source specifies the location of an audio or video resource.

Usage information

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

Rendering expectations

When multiple <media-source> elements are present, the user agent evaluates them in document order and selects the first resource that can be played.

Attributes

The following attributes are available on this element: [Universal attribute group](#) and the attributes defined below.

@href

Specifies the URI of the media resource.

@keyref

Specifies a key reference to the media resource.

@type

Specifies the MIME type of the media resource.

Example

See [audio](#) and [video](#).

<media-track>

Media track settings specify information about supplemental text-based data, for example, a user label, 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 \(WebVTT\)](#).

The content of the `<media-track>` element

Attributes

The following attributes are available on this element: [Universal attribute group](#) and the attributes defined below.

@href

Specifies the URI of the track resource.

@keyref

Specifies a key reference to the track resource.

@kind

Specifies the usage for the track resource. This attribute is modeled on the `@kind` attribute on the HTML5 `<track>` element, as described by the [HTML specification, WHATWG version](#). 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.

@srclang

Specifies the language of the track resource.

Example

See [video](#).

`<video>`

A video is a recording of moving visual images.

Usage information

The `<video>` element is modeled on the HTML5 `<video>` element.

A video resource can be referenced by @href, @keyref, and nested <media-source> elements.

Behaviors such as auto-playing, looping, and muting are determined by attributes. When not specified, the default behavior is determined by the user agent that is used to present the media.

Rendering expectations

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

Processors **SHOULD** scale the video resource 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 resource cannot be rendered in a meaningful way, processors **SHOULD** render the contents of the <fallback> element, if it is present.

Attributes

The following attributes are available on this element: [Universal attribute group](#) and the attributes defined below.

@autoplay

Specifies whether the resource automatically plays when it is presented. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@controls

Specifies whether the presentation of the resource includes user interface controls. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

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

@href

Specifies the absolute or relative URI of the video resource. If @href is specified, specify @type also.

@keyref

Specifies a key reference to the video resource.

@loop

Specifies whether the resource loops when played. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@muted

Specifies whether the resource is muted. The following values are recognized: "true", "false", and "-dita-use-conref-target ". The default value is "true".

@poster

Specifies the absolute or relative URI of the image that is rendered before video playback begins.

@posterkeyref

Specifies a key reference for the poster image.

@tabindex

Specifies whether the video resource can be focused and where it participates in sequential keyboard navigation. See [@tabindex](#) in the HTML specification (WHATWG version).

@type

Specifies the MIME type for the video resource. This attribute enables processors to avoid loading unsupported resources. If `@type` is not specified and `@keyref` is specified, the effective type for the key named by the `@keyref` attribute is used as the value. If an explicit `@type` is not specified on either the `<video>` element or key definition, processors can use other means, such the URI file extension, to determine the effective MIME type of the video.

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

Examples

Figure 8: A <video> element that uses direct addressing

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

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

Figure 9: A <video> element that uses indirect addressing

In the following code sample, the video resource is addressed using a key reference:

```
<video keyref="video"/>
```

Both the URI and the MIME type are specified on the key definition:

```
<keydef keys="video" href="video.mp4" type="video/mp4"/>
```

Figure 10: A <video> element with multiple formats

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

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

Figure 11: Example of a <video> element with multiple formats and multilingual subtitles

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 height="300px"
        loop="false"
        muted="false">
```

```
        poster="demol-video-poster"
        width="400px">
<desc>A video illustrating this procedure.</desc>
<fallback>
  <image href="video-not-available.png">
    <alt>This video cannot be displayed.</alt>
  </image>
</fallback>
<!-- Multiple formats, referenced via key. The key definition
      specifies both the URI and the MIME type -->
<media-source keyref="demol-video-mp4"/>
<media-source keyref="demol-video-ogg"/>
<media-source keyref="demol-video-webm"/>
<!-- Subtitle tracks in English, French and German.
      Each key definition provides a URI and sets type="subtitles". -->
<media-track srclang="en" keyref="demol-video-subtitles-en"/>
<media-track srclang="fr" keyref="demol-video-subtitles-fr"/>
<media-track srclang="de" keyref="demol-video-subtitles-de"/>
</video>
```

Index

M

multimedia elements

`<audio>` [12](#)

`<media-source>` [14](#)

`<media-track>` [14](#)

`<video>` [15](#)