INTERNATIONAL STANDARD

ISO/IEC 23001-15

First edition
2019-12

Information technology — MPEG systems technologies —

Part 15:
Carriage of web resources in ISOBMFF

Technologies de l'information — Technologies des systèmes MPEG —

Partie 15: Transport de ressources web au format ISOBMFF

Reference number
ISO/IEC 23001-15:2019(E)

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

Published in Switzerland

Contents

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23001 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

ISO/IEC 14496-12 specifies a format for the storage of timed resources such as media streams as well as of resources for which no timed stream structure exists, or when the timed stream structure does not need to be exposed.

This document specifies the use of ISO/IEC 14496-12 tools for the storage and delivery of web data. The specified storage is designed to enable enriching audio/video content, as well as audio-only content, with synchronized, animated, interactive web data, including overlays.

Information technology — MPEG systems technologies —

Part 15: Carriage of web resources in ISOBMFF

1 Scope

This document specifies how the format defined in ISO/IEC 14496-12 can be used to store web resources (HTML, JavaScript, CSS, etc.) and defines brands to identify files conforming to this document. It also specifies hypothetical processing for how these files can be consumed by web browsers.

The specified storage enables the delivery of synchronized media and web resources as supported by ISO/IEC 14496-12: file download, progressive file download, streaming, broadcast, etc.

This document also defines how to signal required web capabilities to process the files. This is done in a way that web profiles defined by other organizations can be signalled in a dedicated box, e.g. the MIME Box, similarly to how it is done in ISO/IEC 14496-30.

This document does not define any profiles for web data, only their carriage.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO Base Media file format*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14496-12 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1.1
media player
entity capable of parsing an ISOBMFF file and processing the content of its items and tracks

3.1.2
timed web resource
web resource that has a timing structure exposed in the file format for the purpose of time-driven processing such as seeking or synchronization with other tracks

3.1.3
web engine
part of a web browser in charge of processing web resources and of rendering them

3.1.4
web resource
data processed by a web engine

3.2 Abbreviated terms

CSS	cascading style sheets
HTML	Hypertext Markup Language
ISOBFF	ISO base media file format
JS	JavaScript

4 Hypothetical processing

4.1 Overview

This Clause describes two general hypothetical processing models and corresponding receiver implementations to consume ISOBMFF files conforming to Clause 5, as well as detailed hypothetical processing of items and samples carried in the file.

4.2 General processing models

The first model applies when a media player integrates a web engine. In that case, the ISOBMFF file is processed first by the media player. Track samples and items are extracted from the file. Track samples are processed synchronously (see 4.4) while items are processed when needed (see 4.3). Both types of ISOBMFF constructs carry data that is processed by instantiating a web engine. The video decoding and web engine rendering are then combined, possibly synchronously, for final display. It is depicted in Figure 1.

NOTE Frame accurate synchronization is a goal but can be hard to achieve as instantaneous processing of the sample web content can be challenging due to processing loads. The current model guarantees synchronization at the decoding side and gives the intention to the application that presentation will be synchronized as well, but this is left to the implementation. At this point, exact processing is handled by the implementation and decode-ahead processing could be needed.

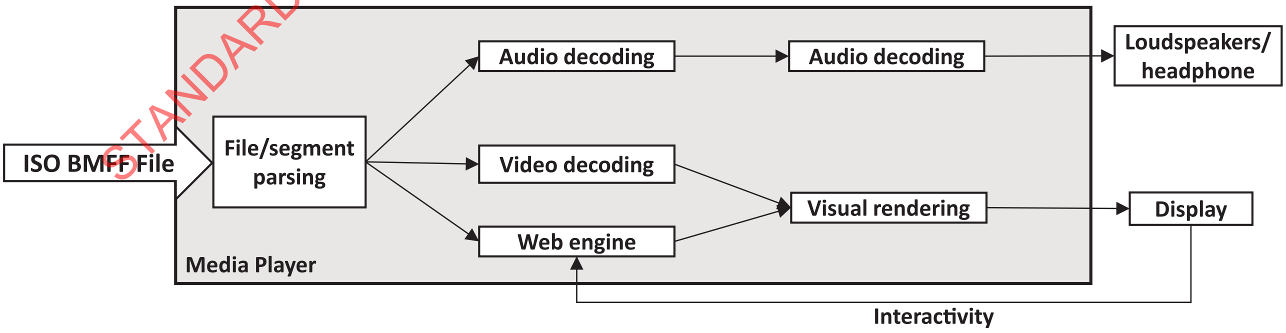

Figure 1 — Hypothetical processing model when an ISOBMFF file carrying web data is passed to a media player integrating a web engine

The second model, depicted in Figure 2, applies when the web engine is initiated prior to the media player, by loading some web content (e.g. an HTML page) and when that content loads an ISOBMFF file (e.g. using a video or audio tag) also containing web content.

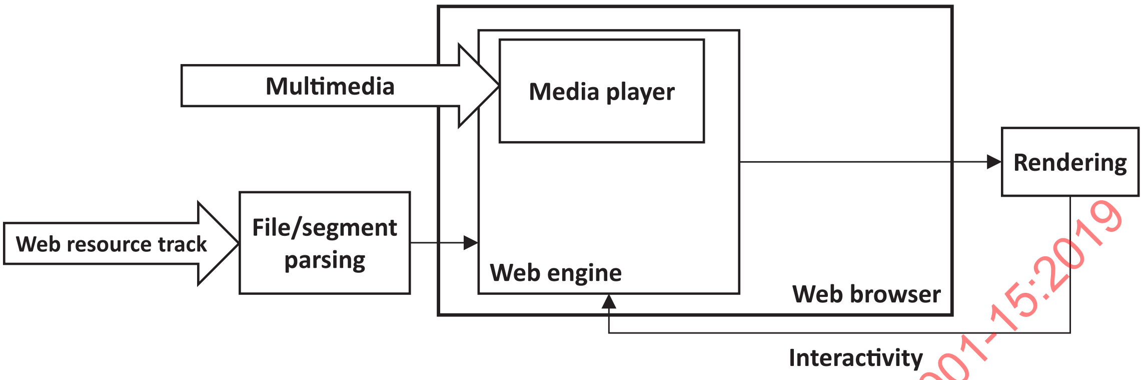

Figure 2 — Hypothetical processing model for web browser-based playback

4.3 Item processing and caching

In both models, resources stored as items in the ISOBMFF file are extracted from the file and placed in the cache of the web engine, as if they had been downloaded from a URL given by the composition of the item's `item_name` field (effectively, a relative URL) with the absolute URL of the containing ISOBMFF file, as defined in ISO/IEC 14496-12.

NOTE Item names are restricted to be relative URLs. Therefore, all resources in the ISOBMFF file come from the same origin as the ISOBMFF file they are contained in.

EXAMPLE If the base URL of the ISOBMFF file is http://example.org/path/file.mp4 and the `item_name` is foo.html, the item content is placed in the cache as if it had been downloaded from http://example.org/path/foo.html.

These resources are then available for the web engine (e.g. through the `XmlHttpRequest` or Fetch APIs). This applies even in the case of streaming using ISOBMFF fragments, when items are delivered progressively, i.e. when `MetaBoxes` are delivered in segments.

4.4 Timed sample processing

For all track types defined in this document, the following processing model is expected.

A track sample becomes active at the presentation time of that sample. The exact processing of an active sample depends on the track type, as described in 5.2. Concrete implementations might have to take into account processing delays to achieve synchronization and might have to pre-fetch resources that are not embedded in the file. When the sample becomes inactive, i.e. when the next sample becomes active, the processing of the sample that was previously active stops. And, for all track types, when an edit list of type 'empty' is applied, no sample is active.

The data carried in a media sample is valid for the entire duration of the sample. However, samples may contain internal timing that makes their presentation evolve over their duration. As a consequence, when seeking at the beginning or in the middle of the sample active period, the result might not be the same, even if the data fed to the decoder is the same.

For all tracks defined in this document, if a sample has its `sample_has_redundancy` flag set, it is expected that that sample would only be made available by the ISOBMFF parser to the web engine if the processing of the file starts with this sample. Otherwise, it is expected that the sample be ignored, and that processing of the current sample is continued for a duration equal to the duration of the ignored sample, as defined in ISO/IEC 14496-12.

5 Carriage of web data in ISOBMFF

5.1 Overview

This Clause specifies how tracks (respectively items) can be used to store timed (respectively untimed) web resources in ISOBMFF files.

5.2 Timed web resources

5.2.1 Overview

Storage of timed web resources can be done using tracks when timed presentation is desired, in particular when operations such as seeking, fast forwarding or trick play are needed.

Several types of tracks are defined in this document, for different purposes:

- Tracks specified in 5.2.3.1 are made of samples whose content is conformant to W3C HTML. These tracks are meant to be used as overlay, when playing other media tracks. Samples in these tracks may embed data conformant to other specifications than W3C HTML, as permitted by W3C HTML, such as JavaScript or CSS. Such embedding may be needed in an environment where single-track files are used, such as CMAF. Examples of usages of this type of track are provided in A.2.
- Tracks specified in 5.2.3.2 are made of samples whose content is conformant to W3C WebVTT carrying metadata. An example of usage of this type of track is given in A.3.
- Tracks specified in 5.2.3.3 are made of samples whose content is text content, without restriction on the format, meant to be used by a web application. An example of usage of this type of track is given in A.4.

Some of the track types specified below, specifically in 5.2.3.2 and 5.2.3.3 rely on a webpage being previously loaded to provide the context for the track content. For those types of tracks, the file should embed the associate webpage and its resources as items as specified in 5.3.2.

A single file may contain multiple tracks defined in this document. The processing of some tracks may be required by some other tracks. The selection of which sets of independent tracks to play is implementation specific. It may be done based on metadata also given in the file (profile capabilities, language information) or based on user input (e.g. through a graphical user interface).

Tracks carrying web resources may use short duration samples, if the changes are frequent, or long duration samples. When long duration samples are used, alternatives to using a track may be considered such as the progressive delivery of items specified in 5.3.

5.2.2 Track layout

Unless specified by an embedding environment (e.g. by an HTML page delivered out-of-band or delivered in-band and identified as the file entry point as specified in 5.3), the layout of tracks, including of overlay HTML tracks and images, is specified using the `TrackHeaderBox` of the different tracks.

NOTE This behaviour is similar to the one specified in ISO/IEC 14496-30.

5.2.3 Track definitions

5.2.3.1 Overlay HTML tracks

5.2.3.1.1 Overview

Images are important web resources. Image sequence tracks, as specified in ISO/IEC 14496-12 or ISO/IEC 23008-12, can be used to provide image overlays. Overlay HTML tracks as specified in this subclause may also be used to provide advanced overlays, with or without interactivity, including image overlays, when the HTML content reference images (either in tracks or as items).

5.2.3.1.2 Content signalling

The brand 'htmt' is used to signal the presence of tracks with the following constraints:

- The track handler type shall be 'text'.
- The sample entry format shall be 'stxt' and:
- its mime_format field shall be set to 'text/html' or 'application/xhtml+xml',
- its content_encoding field shall contain either an empty string or a value allowed in HTTP's Content-Encoding header.
- The content of each sample shall be compliant to HTML. Each sample shall be marked as a sync sample. Samples may have the sample_has_redundancy flag set to 1, in which case ISO/IEC 14496-12 processing is applied as discussed in 4.4.

5.2.3.1.3 Processing model

According to the hypothetical model defined in this document, when sample becomes active, the media player loads the sample data into the web engine and the result of the web engine rendering is presented in the region defined by the width and height of the track header box. When the sample becomes inactive, presentation of the result of the web engine rendering is stopped.

Also according to the model, the passing of the sample data to the web engine is equivalent to the navigation to the document contained in the sample data. The exact processing by web engines is implementation specific, but if the navigation approach is used, it should be avoided that the information is treated as user navigation and for example captured in the browser history.

5.2.3.2 WebVTT metadata tracks

5.2.3.2.1 Overview

This type of track is meant to be used when some web content is loaded first into the web engine and if that content refers (via a <track> element or equivalent JavaScript API) to this ISOBMFF track.

5.2.3.2.2 Content signalling

The brand 'hvtt' is used to signal the presence of tracks with the following constraints:

- The track handler type shall be 'meta'.
- The sample entry format shall be 'wvtt', as specified in ISO/IEC 14496-30 and:
  - its TextConfigBox content shall be "WEBVTT\r\nkind: metadata\r\n".
  - the KindBox shall be present in a UserDataBox in the TrackBox with the following values:
    - schemeURI: about:html-kind, as specified in HTML[1)]

1) https://html.spec.whatwg.org/multipage/media.html#identifying-a-track-kind-through-a-url

  - value: metadata, as specified in HTML[2]

- The content of each sample shall be compliant with WebVTT metadata cues. Each shall be marked as a sync sample. Samples may have the sample_has_redundancy flag set to 1, in which case ISO/IEC 14496-12 processing is applied as discussed in 4.4.

5.2.3.2.3 Processing model

According to the hypothetical model defined in this document, when the sample becomes active, the sample data is made available to the WebVTT parser and as consequence becomes available in the JavaScript context in the form of a TextTrackCue as specified in HTML. When the sample becomes inactive, the associated TextTrackCue is removed from the list of active cues.

5.2.3.3 Web application data tracks

5.2.3.3.1 Overview

This type of track is meant to be used to carry timed content to be consumed by a web application. The exact content of the sample data is dependent on the web application associated with it.

5.2.3.3.2 Content signalling

The brand 'dhtm' is used to signal the presence of tracks with the following constraints:

- The track handler type shall be 'meta'.
- The sample entry format shall be 'stxt' and:
  - its mime_format field shall be set to a text-based MIME type,
  - its content_encoding field shall contain either an empty string or a value allowed in HTTP's Content-Encoding header.
- The content of each sample shall be text content, but is otherwise non-restricted. In particular, it could contain XML content, JSON content or else. Samples may have the sample_has_redundancy flag set to 1, in which case ISO/IEC 14496-12 processing is applied as discussed in 4.4.

NOTE Unlike other track types defined in this document, samples in this track type do not have to be marked as sync samples. They can be marked as non-sync samples.

5.2.3.3.3 Processing model

According to the hypothetical model defined in this document, when the sample becomes active, the sample data is made available at the relative URL “<file.mp4>#trackID=<x>”, where ‘<file.mp4>’ designates the ISOBMFF container (and can even be omitted), and ‘<x>’ is replaced by the trackID of the track. The sample data is available at that URL until the sample remains active.

5.3 Non-timed web resources

5.3.1 Overview

Web resources can be stored as items, when there is no specific time associated to the loading of that resource or when the track storage is not appropriate.

As defined in ISOBMFF, items are declared in a MetaBox which may be present in the movie header or in movie fragment headers.

[2] https://html.spec.whatwg.org/multipage/media.html#text-track-model

Items carrying web resources may serve either as an entry point to the file (e.g. as determined by the application using the ISOBMFF file or when the ISOBMFF is loaded with a URL with fragment identifier identifying that item) or as secondary content loaded by either other items or track samples.

An example of such storage is given in A.1.

5.3.2 Web Items

The brand `'htmi'` may be used to signal the use of a `MetaBox` with the following constraints:

- It shall be present at the file level.
- It shall use a `HandlerBox` with the `handler_type` set to `'html'`.
- It shall contain a `PrimaryItemBox` which declares as primary item a resource of type `'text/html'`, or `'application/xhtml+xml'`.
- It shall not use any `DataInformationBox`, `ItemProtectionBox` or `IPMPControlBox`.
- It shall use a `ItemInformationBox` with the following constraints:
  - its version is either 0 or 1;
  - each item is described by an `ItemInfoEntry` with the following constraints:
    - its version is set to 0;
    - its `item_protection_index` is set to 0;
    - if the item is referred to by a URL in the content of another item, its `item_name` is equal to that URL.
- It shall use an `ItemLocationBox` with the following constraints:
  - its version is set to 1 or 2;
  - each item is described by an entry and values 0, 1 or 2 may be used for the construction method.
- It may use any other boxes (such as `ItemReferenceBox`) not explicitly excluded above.

5.4 URLs to web resources embedded in ISOBMFF files

If web resources are embedded in an ISOBMFF file (either in track samples or items as described above), and if any of these resources contains URLs to other resources also embedded in the same ISOBMFF file (including to the ISOBMFF file itself (e.g. in a video tag)), if the file declares in its FileTypeBox any of the brand defined in this document, the following constraints apply on the file:

- A `MetaBox` shall be present at the file level with the following constraints:
  - For each URL to a resource embedded in the ISOBMFF file, the `item_name` for the item corresponding to that resource shall be set such that the resource is accessible assuming the processing described in 4.3.
  - Relative URLs shall be used. The Base URL of the including ISOBMFF resource defines the Base URL for reference resolution following RFC 3986.
  - If there is a need for an item with a known name, that represents the entire containing file, then an item shall be constructed using:
    - a self-contained data reference;
    - an item body with `offset` set to 0 and `length` indicating the size of the entire file;

- an item name that is the desired known name.

This effectively constructs an alias to the entire original file.

Annex A
(informative)

Examples

A.1 Web item examples

A.1.1 Embedded HTML entry point playing the video content

Assuming web content composed of the following HTML document:

```
<!doctype html>
<html>
    <body>
        <video src="file.mp4"></video>
    </body>
</html>
```

An ISOBMFF file named ‘file.mp4’ storing this resource conformant to this document could be structured as follows:

- track 1: audio track
- track 2: video track
- one file level MetaBox with the following values:
  - handler: html
  - primary item: 1
  - items:
    - id: 1, content_type: text/html, name: (any name), location: offset O1, length L1, item content is the HTML file
    - id 2: content_type: video/mp4, name: file.mp4, location: 0, offset: entire file length, item content is the entire ISOBMFF file

If played by a media player not capable of processing the MetaBox with a handler of type ‘html’, only the audio-video tracks are played.

If played by a media player capable of processing files conformant to this document, the following behaviour is expected: first the HTML page is extracted from the primary item and loaded into the web engine; then the web engine when processing the video tag will, requesting playback of the audio/video file, therefore, the audio/video tracks are played within the HTML page.

A.1.2 Embedded HTML entry point playing the video content with additional JavaScript embedded static file

Assuming web content composed of the following HTML document and of a JavaScript file (named file.js whose content is not provided here):

```
<!doctype html>
<html>
    <head>
        <script src="file.js">
    </head>
    <body>
        <video src="file.mp4"></video>
    </body>
</html>
```

An ISOBMFF file named ‘file.mp4’ storing this resource conformant to this document could be as follows:

- track 1: audio track
- track 2: video track
- one file level MetaBox with the following
  - handler: html
  - primary item: 1
  - items
    - id: 1, content_type: text/html, name: (any name or no name), location: offset O1, length L1
    - id 2: content_type: video/mp4, name: ‘.’, location: 0, offset: entire file length
    - id 3: content_type: application/javascript, name: file.js, location: X2, offset: L2

A.1.3 Non-embedded HTML page launches entry point HTML stored in an ISOBMFF file

Assuming the web content is the following HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <script>
     function onload() {
        var videoURL="video.mp4";
        // some logic fetch the video file and find the entryPageURL in MetaBox
        document.getElementById('content').src = entryPageURL;
     }
    </script>
  </head>
  <body onLoad="onload()">
        <!-- The web content embedded in the video file is presented in the iframe -->
        <iframe id="content" width="1280" height="960" allowfullscreen/>
  </body>
</html>
```

In this example, the JavaScript code fetches the MP4 file (video.mp4) and extracts the HMTL entry point in that file, and then links it with the src attribute of the iframe.

The ISOBMFF file named 'video.mp4' can be structured as follows:

- track 1: audio track
- track 2: video track
- one top level `MetaBox` with the following
  - handler: html
  - primary item: 1
  - items
    - id: 1, content_type: text/html, name: (any name), location: offset O1, length L1
    - id 2: content_type: video/mp4, name: '.', location: 0, offset: entire file length

The HTML content in the `MetaBox` can be as follows:

```
<!doctype html>
<html>
    <body>
        <video src="video.mp4"></video>
    </body>
</html>
```

A.1.4 Delivery of non-timed resources required by web tracks

Assuming an HTML overlay track, as specified in 5.2.3.1, with multiple samples as follows:

Sample 1: presentation time: 0, sync sample

```
<!doctype html>
<html>
    <body>
        <img src="file1.jpg"></img>
    </body>
</html>
```

Sample 2: presentation time: 1 s, sync sample

```
<!doctype html>
<html>
    <body>
        <img src="file2.jpg"></img>
    </body>
</html>
```

Sample 3: presentation time: 5 s, sync sample

```
<!doctype html>
<html>
    <body>
        <img src="file3.jpg"></img>
    </body>
</html>
```

The different image files (file1.jpg, file2.jpg, file3.jpg) required by the HTML documents can be stored in one top-level MetaBox as follows:

- handler: pict
- primary item: 1
- items
  - id 1: content_type: image/jpeg, name: file1.jpg, location: offset: O1, length L1
  - id 2: content_type: image/jpeg, name: file2.jpg, location: offset: O2, length: L2
  - id 3: content_type: image/jpeg, name: file3.jpg, location: offset: O3, length: L3

The content of the MetaBox may be split into several MetaBoxes. Assuming the file is segmented with a segment length of 1s, the composition of the segments would be:

Segment 1:

- sample 1
- a MetaBox with item 1 only

Segment 2:

- sample 2
- a MetaBox with item 2 only

Segment 3 / Segment 4 / Segment 5:

- sample 2, `sample_has_redundancy` set to 1
- a MetaBox with item 2 only

Segment 6:

- sample 3
- a MetaBox with item 3 only

The track can be marked as dependent on the item 1, 2 and 3 using 'dpnd' track references. Samples in the track may also be marked with a SampleToMetadataItemEntry sample group 'stmi' to indicate the sample dependency to the items.

A.2 HTML overlay tracks

A.2.1 Track content

As an example, the content of an HTML overlay track could be as follows:

Sample: presentation time: 0, sync sample

```
<!doctype html>
<html>
    <body>
        Some text
    </body>
</html>
```

Sample: presentation time: 10 s, sync sample

```
<!doctype html>
<html>
    <body>
        Some other text with graphics <svg>…</svg>
    </body>
</html>
```

A.2.2 Track layout using HTML

Assuming the file 'file.mp4' contains only audio/video and that the file 'overlay.mp4' contains only a HTML overlay track whose content is given in A.2.1, the following HTML file could be provided to indicate that the overlay track is meant to be displayed as a subtitle track.

```
<video src="file.mp4">
<track src="overlay.mp4" kind="metadata">
</video>
```

Similarly, assuming that the file 'file.mp4' contains an audio track (id 1), a video track (id 2), and the overlay HTML track (id 3), the HTML providing the track layout could be the following:

```
<video src="file.mp4">
<track src="file.mp4#trackID=3" kind="metadata">
</video>
```

NOTE The HTML could be embedded in the file as an item following the example in A.1.