W3C TTML Profiles for Internet Media Subtitles and Captions

IMSC

End-to-End Internet Subtitles and Captions

Pierre-Anthony Lemieux, Sandflow Consulting (pal@sandflow.com)
Supported by MovieLabs
IMSC

Developed by the W3C Timed Text Working Group (TTWG)

Application of **TTML** for subtitles and captions

Reduces fragmentation by bringing together multiple profiles of TTML

Focal point for Internet subtitles and captions

Evolve with worldwide requirements
## Working with other formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMPTE-TT</td>
<td>Likely no conversion necessary to IMSC1</td>
</tr>
<tr>
<td>CEA 608</td>
<td>SMPTE RP 2052-10</td>
</tr>
<tr>
<td>CEA 708</td>
<td>SMPTE RP 2052-11</td>
</tr>
<tr>
<td>EBU-TT-D</td>
<td>No conversion necessary to IMSC1</td>
</tr>
<tr>
<td>EBU STL</td>
<td>Via EBU-TT-D (<a href="https://www.ebu.ch/publications/3360">EBU Tech 3360</a>)</td>
</tr>
<tr>
<td>WebVTT</td>
<td><a href="https://www.w3.org/TR/webvtt/">Draft mapping</a> developed by the TTWG</td>
</tr>
</tbody>
</table>
# Adoption

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMPTE ST 2067-2</td>
<td>Interoperable Master Format (IMF)</td>
</tr>
<tr>
<td>ISO 23000-19</td>
<td>Common media application format (CMAF) for segmented media</td>
</tr>
<tr>
<td>DVB A174</td>
<td>Digital Video Broadcasting (DVB); TTML Subtitling Systems</td>
</tr>
<tr>
<td>ATSC A/343</td>
<td>ATSC Standard: Captions and Subtitles</td>
</tr>
<tr>
<td>CTA WAVE</td>
<td>Consumer Technology Association: Web Application Video Ecosystem</td>
</tr>
</tbody>
</table>
End-to-End

Mastering → Library → Online Service → CDN → Consumer Device

- IMF (SMPTE ST 2067)
- MXF (SMPTE ST 377-1)
- JPEG 2000, etc...
- PCM
- CMAF (ISO 23000-19) + DASH (ISO 23009) | HLS
- ISO BMFF (ISO 14496-30)
- AVC, etc.
- AAC, etc.

IMSC

Twentieth Century Fox, etc.

iOS, web browser, etc.
Demo

http://subtitling.irt.de/cmaf/
Case Study: IMF

Component-based master format (SMPTE ST 2067-2)
Basics

Regions defined relative to a root container

Text and images flow into regions at specified time coordinates, e.g. HH:MM:SS.fraction

Time coordinates are offsets from T=0s

- Not a timecode timestamp
- Not tied to video timecode
Anatomy of an IMSC1 Document

<?xml version="1.0" encoding="UTF-8"?>
<tt xml:lang="en" xmlns="http://www.w3.org/ns/ttml" ...>
<head>
  <styling>
    <style xml:id="baseStyle" tts:color="white" tts:textAlign="center"/>
    <style xml:id="blackBackground" tts:backgroundColor="black"/>
    <style xml:id="greenBackground" tts:backgroundColor="green"/>
    <style xml:id="withLinePadding" ebutts:linePadding="0.5c"/>
  </styling>
  <layout>
    <region xml:id="area1" tts:origin="5% 10%" tts:extent="90% 20%" tts:displayAlign="center"/>
    <region xml:id="area2" tts:origin="5% 70%" tts:extent="90% 20%" tts:displayAlign="center"/>
  </layout>
</head>
<body>
  <div style="baseStyle">
    <p region="area1" begin="00:00:01" end="00:00:09">
      <span style="greenBackground">Centered text on two lines<br/>without padding.</span>
    </p>
    <p region="area2" style="withLinePadding" begin="00:00:01" end="00:00:09">
      <span style="blackBackground">Centered text on two lines<br/>with padding.</span>
    </p>
  </div>
</body>
</tt>
Flexible styles and writing modes

The last words must not be italic.

The quick brown fox jumps over the lazy dog.

Centered text on two lines without padding.

Centered text on two lines with padding.

This text has a red, two-pixel outline.

The last word in this caption is underlined.
Test Suite

**FillLineGap001.ttml**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<tt xmlns="http://www.w3.org/ns/ttml"
...
<body>
  <div>
    <p xml:id="subtitle1" region="bottom"
      begin="00:00:00.000" end="00:00:30.000"
      style="paragraphStyle">
      <span style="spanStyle">##Line gaps##</span>
    </p>
    ...</div>
  </body>
</tt>
```

**Exemplar Render**

![Exemplar Render](https://github.com/w3c/imsc-tests)
## A Few Open Source Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imscJS</td>
<td>JavaScript library for rendering IMSC documents to HTML5</td>
</tr>
<tr>
<td>Timed Text Toolkit (ttt)</td>
<td>Java-based TTML renderer and validator</td>
</tr>
<tr>
<td>MP4Box</td>
<td>ISO BMFF multiplexer</td>
</tr>
<tr>
<td>dash.js</td>
<td>Reference DASH web player</td>
</tr>
<tr>
<td>asdcplib</td>
<td>Wraps IMSC in MXF</td>
</tr>
</tbody>
</table>

Many other projects with some IMSC compatibility, e.g. Shaka Player, Exo Player…
What is imscJS?

Open source JavaScript library
Renders IMSC documents to HTML5
Implements IMSC 1.0.1 today, and IMSC 1.1 soon
Used by dash.js (reference DASH player)
Supported by MovieLabs and Netflix

https://github.com/sandflow/imscJS
Demos

Sample imscJS web application: http://sandflow.com/imsc1proc/index.html


Browser support for subtitle and caption stylistic features is not perfect.
IMSC Status Roadmap

IMSC 1 Recommendation [April 2016]

IMSC 1.0.1 Recommendation [April 2018]

IMSC 1.1 Candidate Recommendation [May 2018]

Publication of IMSC 1.1 Recommendation planned for October 2018
IMSC 1.1

Based on TTML 2

Superset of IMSC 1.0.1

- IMSC 1.0.1 document is a valid IMSC 1.1 document
- IMSC 1.1 processor presents an IMSC 1.0.1 document as it would have been presented by an IMSC 1.0.1 processor
- A few deprecated features
- A number of new features...

https://github.com/w3c/imsc-vnext-reqs
Ruby
Tate-Chu-Yoko
Emphasis Marks

花よりだんご
花よりだんご
Shadow

I serve it with greens in those shadowy scenes,
And I use it for striking a light:
Stereoscopic 3D

tts:disparity sets binocular disparity between renderings of a region

- Positive disparity \(\rightarrow\) perceived behind the plane of the display
- Negative disparity \(\rightarrow\) perceived in front of the plane of the display

Similar to SMPTE ST 428-7 (D-Cinema) and CEA 708.1

Ignored when not rendering onto a stereoscopic image pair

\[
\begin{align*}
\text{Left eye} & \quad \text{Right eye} \\
\text{hello} & \quad \text{hello} \\
\text{disparity} < 0 & \quad \text{disparity}
\end{align*}
\]
Two options

- Mapping of SDR RGB colors to HDR presentations
- Carry PQ images in PNG

<table>
<thead>
<tr>
<th>Map sRGB onto PQ</th>
<th>Map sRGB onto HLG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author-supplied luminance gain (TTML2 tts:luminanceGain and Annex Q.1)</td>
<td>Fixed recommended mapping (TTML2 Annex Q.2)</td>
</tr>
<tr>
<td>Ignored if compositing onto SDR image</td>
<td></td>
</tr>
</tbody>
</table>
Mapping SDR RGB to PQ image

sRGB × 80 nits × \text{tts:luminanceGain} \ \text{Inverse PQ OETF} \ \text{PQ}

\text{rgb}(218, 165, 32) \ \text{tts:luminanceGain} = \text{"1.5"} \ \Rightarrow \ \text{absoluteRGB}(82.37, 42.21, 0.82)
Mapping sRGB to HLG image

sRGB → Linear RGB → \times 0.265 → HLG OETF → HLG
PQ in PNG

Using the ITU BT.2100 PQ EOTF with the PNG Format (W3C WG Note)

Uses the existing iCCP chunk profile name to signal BT.2100 PQ images

Graceful processing by legacy decoders with fallback

- gAMA chunk
- cHRM chunk
- embedded ICC profile

Code and examples at https://github.com/sandflow/hdr4png
Thank you!