Internet Captioning - Implications of the Multi-platform, Multi-Display Ecosystem

Welcome

SMPTE Monthly Webcast Sponsors

- Thank you to our sponsors for their generous support of SMPTE and the SMPTE Professional Development Academy.
SMPTE Monthly Webcasts:
Internet Captioning - Implications of the Multi-platform, Multi-Display Ecosystem

Your Host

Joel E. Welch
Director of Professional Development
SMPTE

SMPTE Monthly Webcasts

• Series of monthly 1-hour online, interactive webcasts covering a variety of technical topics
• Free professional development benefit for SMPTE members
• Sessions are recorded for member viewing convenience.
Housekeeping

- Please use the chat box to submit questions to the speaker
- A PDF of slides used during today’s webcast will be provided in exchange for your feedback
  - Near the end of the presentation, you will be provided a link to the feedback form
  - Once your feedback is submitted, you will automatically be directed to the PDF for downloading

Today’s Guest Speaker

Jason Livingston
Developer and product manager
CPC Closed Captioning
Background

- New FCC regulations require closed captions from TV broadcasts to be available when these videos are delivered by Internet Protocol (IP), such as on the web and mobile devices.

- SMPTE created a new specification called SMPTE Timed Text (2052) to address the challenges of bringing closed captions from broadcast TV to IP video.

- Why is Closed Captioning required?
  - It’s the law
  - About 20% of US households use captions
  - More than 48 million Americans have hearing loss
  - For noisy environments (gyms, restaurants, airports, etc.)

FCC Requirements for IP Captioning

- IP closed captioning must substantially replicate the look and feel of TV broadcast captions, including formatting and positioning

- IP video players are required to implement the same user controls as digital TVs (caption font size, color, etc. are user adjustable)

- SMPTE Timed Text is specifically singled out as a “safe harbor” format

- Loosely interpreted, this means that if you accept SMPTE TT on the input side, deliver SMPTE TT on the output side, and adhere to the SMPTE TT specification, you are considered to be in compliance even if problems occur
FCC Deadlines for IP Captioning

The IP closed captioning rules apply to non-exempt full-length programming and will be implemented according to the following schedule:

- Pre-recorded programming that is not edited for Internet distribution must be captioned if it is shown on television with captions on or after **September 30, 2012**.

- Live and near-live programming must be captioned if it is shown on television with captions on or after **March 30, 2013**. Near-live programming is video programming that is performed and recorded less than 24 hours prior to the time it was first shown on television.

- Pre-recorded programming that is edited for Internet distribution must be captioned if it is shown on television with captions on or after **September 30, 2013**.

“Why is CC so complicated? It’s just text and time codes right?”

Design decisions and limitations of 1970s and 80s technology continue to affect the development of the latest standards:

- Low bandwidth analog transmission
- Must be nearly stateless (can jump into the stream at any point and start decoding)
- Decoder can be implemented with only a few KB of memory
- Support for (limited) text formatting and positioning around the screen
- Character set that can accommodate most Roman alphabet-based languages
- Very limited processing power and memory available for decoders
- Must not interfere with existing NTSC receivers in use
Examples of CEA-608 Caption Features

- Vertical & Horizontal Positioning
- Justification
- Split captions
- Roll-up (smooth scrolling)
- Special characters

What about CEA-708?

Despite the many advantages and new features of CEA-708, we are still stuck with the limitations of CEA-608:

- Most caption authoring tools still target the CEA-608 spec only
- Common caption interchange files (SCC, CAP, etc.) are CEA-608 only
- Vast archives of captioned content contain only CEA-608 captions
- Primary language captions must be backwards compatible with CEA-608 (which implies that many new features of CEA-708 cannot be used)
- Many QC hardware and software do not properly display CEA-708 captions if they make use of the enhanced features
- As a result, most broadcast chains continue to use CEA-608 captions
- CEA-608 data is only translated to CEA-708 as the last step of the broadcast chain
Move to IP/web Delivery

Streaming video should replicate broadcast captions as closely as possible, so that people who rely upon them do not get a second class experience.

Prior/Current efforts for web captioning formats:
- SAMI, SMIL, SRT, AMF onTextData - do not support all CEA-608 features
- TTML (DFXP) – many players only support a limited subset (e.g. no positioning controls), so they don’t support all CEA-608 features
- Native CEA-608/708 embedded data - requires decoder support in every player (not trivial), not supported by all video container formats

Move to IP/web Delivery (cont’d)

Desired features for a new caption format:
- "Mezzanine" format - one master file for both broadcast and IP delivery
  - Raw CEA-608/708 data for broadcast TV
  - A more easily processed format for web-based players
- Must work together with existing caption authoring tools and standard practices
- Support both live (real-time) and post (VOD) transmission
- Format agnostic: should work with any video codec/wrapper and streaming system, both as an embedded stream and as a sidecar file format
- Must address all of the FCC and other legal requirements for closed captions
CEA-608 vs. TTML Methodology

Native CEA-608/708 closed caption data is transmitted as a stream of commands and text, at a fixed number of bytes per frame of video. The data is not human readable – it has to be parsed to be understood.

CEA-608 Representation:

<table>
<thead>
<tr>
<th>Frame</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>9420</td>
<td>9476</td>
<td>97a1</td>
<td>9137</td>
<td>e2d5</td>
<td>d349</td>
<td>4380</td>
<td>9137</td>
<td>942c</td>
<td>942f</td>
</tr>
<tr>
<td>Meaning</td>
<td>Pop-on mode</td>
<td>Row 15</td>
<td>TAB</td>
<td>Indent</td>
<td>♬</td>
<td>MU</td>
<td>SI</td>
<td>C</td>
<td>♬</td>
<td>Clear Screen</td>
</tr>
</tbody>
</table>

Contrast with timed text format e.g. TTML, which is a document containing markup and text, and is mostly human readable:

```html
<p begin='00:00:00:09' end='00:00:02:01'>♪ MUSIC ♬</p>
```

Current industry use of SMPTE-TT

- **Authoring**
  - Currently most SMPTE-TT / TTML files are converted from another format like CEA-608 or subtitles, rather than authoring to natively target SMPTE-TT features.

- **Mezzanine**
  - TTML and SMPTE-TT are definitely seeing widespread adoption as an intermediate file format, such as when delivering video to a content distributor, and as a source template for conversions for distribution.

- **Distribution**
  - TTML (DFXP) has been in use by many online video providers, but typically in a limited role without full CEA-608 features (not FCC compliant for broadcast TV content)
  - Limited use of SMPTE-TT as a delivery format (in other words, the file that is actually transmitted to the end user), mostly in provider-specific custom apps.
So, why is this so hard to do on the web?

Broadcast video vs. Web video

Every station broadcasts the same spec, and every TV receives the same spec.
Broadcast video vs. Web video cont’d

Video Delivery
- HTTP
  - Progressive Download (VOD only)
- Apple HLS
- Adobe HDS
- MS Smooth Streaming
- MPEG DASH
- Flash (RTMP)
- Silverlight
- Other (e.g. IPTV)

Captions Delivery
- Video with embedded CEA-608/708 Data
- Sidecar Caption file
  - SMPTE-TT
  - WebVTT
  - TTML (DFXP)
  - Other (SRT, SCC, etc.)

Multi-display, Multi-device

- Ideally, the same video and captions would work on every device, however that is not yet the case.

- Given that we must deliver multiple formats for now, ideally these formats would be industry standards like HTML5 and SMPTE-TT

- Ideally, these formats would be supported natively by the device

- That’s a lot of “ideally”...
### Support for Embedded CEA-608/708 in Video

<table>
<thead>
<tr>
<th>Browser</th>
<th>Native support Embedded CEA-608/708 Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome (Desktop)</td>
<td>☒</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>☒</td>
</tr>
<tr>
<td>Firefox</td>
<td>☒</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>☒</td>
</tr>
<tr>
<td>Opera</td>
<td>☒</td>
</tr>
<tr>
<td>Safari (Desktop)</td>
<td>☑</td>
</tr>
<tr>
<td>Safari (iOS)</td>
<td>☑</td>
</tr>
</tbody>
</table>

Tested using latest public releases as of 6/1/2013

### HTML5 <track> Support for TTML / SMPTE-TT

<table>
<thead>
<tr>
<th>Browser</th>
<th>Native support TTML (SMPTE-TT / DFXP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome (Desktop)</td>
<td>☒</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>☒</td>
</tr>
<tr>
<td>Firefox</td>
<td>☒</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>☒</td>
</tr>
<tr>
<td>Opera</td>
<td>☒</td>
</tr>
<tr>
<td>Safari (Desktop)</td>
<td>☒</td>
</tr>
<tr>
<td>Safari (iOS)</td>
<td>☒</td>
</tr>
</tbody>
</table>

Tested using latest public releases as of 6/1/2013
HTML5 <track> Support for WebVTT

<table>
<thead>
<tr>
<th></th>
<th>Display WebVTT</th>
<th>CEA-608 Styling</th>
<th>JavaScript API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome (Desktop) 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Firefox</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Opera</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Safari (Desktop)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Safari (iOS)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Tested using latest public releases as of 6/1/2013
1 The ability to recreate most or all of CEA-608 styling/formatting settings
2 Chrome support requires modifying developer flags via chrome://flags

Supplementing Browser’s Support

- Different browsers might not support the HTML5 <track> format you’re using, or may only support a subset of its features

- A JavaScript “polyfill” can fill in for some of the missing features in the browser, e.g.
  - Parsing the caption file specified by <track>
  - Displaying the captions over the HTML5 Video
  - Formatting & Positioning the captions per the caption file

- However...
Supplementing Browser’s Support cont’d

• Many devices do not play the video in the web browser itself, rather they open in a separate video player application

• This application may only access the video track and completely ignores other <track> elements and JavaScript

• Therefore the captions must either be embedded into the video file itself, or the video player device/app must be capable of fetching and displaying a sidecar captions file

Supplementing Browser’s Support cont’d

• A workaround to these issues is to create a custom player application

• This player can support whatever combination(s) of formats you’d like to support

• App must support all mandated FCC features such as caption formatting and TV-style user controls for size, font, etc.

• Will need to be developed and supported for each platform that you want to support
Format Fragmentation

- Different devices require not only different video types, but also different caption types
- Requires the server or CDN to supply the same contents in multiple formats

TTML Fragmentation

- This is one of the biggest problems hurting TTML / SMPTE-TT adoption today
- Many vendors’ SMPTE TT / TTML files are compliant with the spec, but have to be specially tailored for a specific player or delivery mechanism
- Requires vendors to create multiple variants of SMPTE TT files to target different devices
### TTML Fragmentation cont’d

Variety of SMPTE-TT / TTML / DFXP exports available in CPC Software

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crackle Timed Text (.xml)</td>
<td></td>
</tr>
<tr>
<td>Flash Text (.txt)</td>
<td></td>
</tr>
<tr>
<td>Flash XML for Captionate (.xml)</td>
<td></td>
</tr>
<tr>
<td>Flash XML for CS4 [Timed Text DFXP] (.xml)</td>
<td></td>
</tr>
<tr>
<td>iTunes (.itc)</td>
<td></td>
</tr>
<tr>
<td>QuickTime (.txt, .smi)</td>
<td></td>
</tr>
<tr>
<td>QuickTime 608 Closed Captions (FCP/Podcast/Web)</td>
<td></td>
</tr>
<tr>
<td>QuickTime Subtitles Track</td>
<td></td>
</tr>
<tr>
<td>QuickTime Text Track</td>
<td></td>
</tr>
<tr>
<td>RealText (.rt, .smi)</td>
<td></td>
</tr>
<tr>
<td>SMPTE Timed Text (.xml)</td>
<td></td>
</tr>
<tr>
<td>Sony Pictures Timed Text (.xml)</td>
<td></td>
</tr>
<tr>
<td>Timed Text DFXP (.xml)</td>
<td></td>
</tr>
<tr>
<td>WebVTT (.vtt)</td>
<td></td>
</tr>
<tr>
<td>Windows Media SAMI file (.smi)</td>
<td></td>
</tr>
<tr>
<td>YouTube (.srt)</td>
<td></td>
</tr>
</tbody>
</table>

### Challenges specific to Live Broadcast

- Live broadcasts are typically captioned by a real-time court reporter using a stenograph (special keyboard), or by a voice captioner using real-time speech recognition software
  - Real-time captioner feeds CEA-608 data into the broadcast hardware
  - Broadcast hardware then translates the CEA-608 caption data into CEA-708 data as the last step before ATSC encoding/transmission.
First Challenge:
Streaming the video & CC from the content provider to the Internet

• How to deliver the video + captions to the Internet?
  – Not all Content Delivery Networks (CDNs) support real-time reception of closed captioning data
  – Even if they do, what format(s) do they support?
  – Do these formats support all of the FCC mandated features (e.g. caption positioning and formatting)?
  – How do they pass the captions on to the end user?

Live video streaming – Native support

<table>
<thead>
<tr>
<th></th>
<th>Apple HLS</th>
<th>Adobe HDS</th>
<th>MS Smooth</th>
<th>Flash (RTMP)</th>
<th>RTSP</th>
<th>MPEG DASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome (Desktop)</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Firefox</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Opera</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Safari (Desktop)</td>
<td>✓</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Safari (iOS)</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Tested using latest public releases as of 6/1/2013
✓ = Native support  O = Via available plugin (Flash, Silverlight, QuickTime, etc.)
Method 1: In-band transmission of CC data

The video encoder for the IP stream needs to include the CEA-608/708 data from the broadcast feed in the IP stream.

The website or CDN must be able to receive and process this data in real-time, so that it can be sent to the viewers.

Method 2: Out-of-band transmission of CC data

The video encoder for the IP stream needs to extract the CEA-608/708 data from the broadcast feed, and generate a chunked SMPTE 2052 stream in real-time.

These chunked segments would either be embedded into the video data, or delivered as separate files to the player.
Second Challenge:
Displaying the CC in the user’s browser/device

- Ideally all devices will eventually support SMPTE 2052 and/or CEA-608/708, but this is not the case yet.
- If a device does not support your preferred CC format, a workaround is to create a custom player app which is tailored to view your streams:
  - Gives the content provider full control over what caption formats can be supported by the device.
  - Involves more work since each provider must create, test, and maintain a custom app – possibly multiple apps for multiple devices.
  - App must support all mandated FCC features such as caption formatting and user controls.

Second Challenge:
“In an ideal world”

- Universal support for a format like SMPTE TT would allow all devices to play the same video and CC.
Second Challenge (cont’d):
In the real world we currently face

- Native video APIs embedded into many browsers and devices don’t necessarily have closed captioning support, or may support only one particular format
- To support a wide variety of mobile devices, providers need to supply the captions in multiple formats, or the CDN must do real-time conversion

![](diagram1.png)

Second Challenge (cont’d):
Another potential issue

- Many vendors’ SMPTE TT files are compliant with the spec but mutually incompatible
- May require multiple versions of SMPTE TT files to target different devices

![](diagram2.png)
Status of these Challenges

- Why is this so difficult?
  - Requires cooperation between the streaming encoders, CDNs, and playback device manufacturers
  - It is possible for each piece of the puzzle to claim “closed caption support” and yet when put together, the system does not work
  - Each entity has their own timelines

- Current Status as of July 2013:
  - Many streaming encoders do support some kind of CC data, but they may not support the kind your CDN needs
  - Many CDNs support CC for some devices but not others
  - Each browser / mobile device / operating system is at different stages of implementing playback support for different formats

Other challenges: Re-broadcast of Edited Content

- Broadcast videos are often served with different commercials or edited content for the web versions
- Re-captioning these videos from scratch is a costly and time consuming process
- Editing systems do not necessarily preserve the existing closed captions at all, and even if they do, the captions are not automatically conformed to match the edited versions
- Tools now exist which can automatically conform the original captions to match an edited version by using a Edit Decision List (EDL) generated from the editing software
- The conformed captions can then be saved as native CEA-608/708, or converted to SMPTE 2052 or other formats
Other challenges: Legal Rights to the Captions

Some closed captions are copyrighted by a different entity than the owner of the video rights

- Rights for captions created by a 3rd party provider may only cover the original TV broadcast and may not allow for conversion of the caption assets into different formats; or may not allow the captions to be edited.

- It is important for all video content creators and distributors to consider the legal ramifications of having a 3rd party author the captions, especially if that 3rd party maintains rights or restrictions on the captions they provide.

- The National Association for the Deaf (NAD) has testified before the US Copyright Office that closed captions and other accessibility technologies should be exempted from copyright and anti-circumvention measures under copyright law.

Recommended Best Practices

For Captioners

- Although CEA-708 and TTML have many more features and options than CEA-608, captioners targeting the North American TV market should continue to author for the CEA-608 spec for backwards compatibility.

- It is recommended that new post produced content be authored in pop-on mode. Pop-on captions are easier to read for people who rely on captions, and deliver a more consistent and reliable experience across a range of platforms. Roll-up captions are typically lower quality and harder to read, and are harder to represent on a variety of devices, thus should only be used for live captioning.

- Ensure that the legal rights for the captions allow the content providers and distributors to make format conversions, to re-broadcast the captions when the content has been edited and via a wide range of delivery mechanisms, etc.
Recommended Best Practices
For Content Creators

• Authoring the captions in-house can eliminate legal issues and cut down on captioning time and expense

• Request that new caption projects be done in pop-on mode, except for (near) live productions

• Captions should ideally be stored in a separate mezzanine format in addition to being embedded into the video asset (file and/or tape). Re-use and re-purposing of standalone caption files is faster and easier than later extracting the caption data from the video asset.

• Ensure that you have in-house capability to handle caption-related tasks such as editing and conversion, ideally in an automated fashion, even if you outsource the labor intensive parts of the captioning process

Recommended Best Practices
For Content Providers/Distributors

• Depending on the software you have available, you may need to specify that content providers deliver only a limited subset of closed captioning files types; or, with more advanced software you can accept a wider range of file types

• Editing and re-purposing of existing caption assets (e.g. commercial removal and segmentation) can usually be done with at least some degree of automation.

• Accept and deliver SMPTE 2052 and/or native CEA-608/708 caption data to meet the FCC safe harbor regulations
Recommended Best Practices
For Web/Mobile Device Developers

- Support SMPTE 2052 and/or implement a native CEA-608/708 broadcast closed captions decoder to meet the FCC safe harbor regulations

- Watch out for "supported" formats that are not fully supported (e.g. limited subset of TTML supported by many current web players), or don't support required features (e.g. SubRip SRT)

- Ensure all FCC mandated decoder features (caption positioning and formatting, user controls, etc.) are fully implemented

Conclusions

- The original design constraints of CEA-608 continue to affect the development of new closed captioning standards such as SMPTE 2052.

- While hardware and software are still catching up with specifications such as CEA-708 that have been around for years, new video formats and distribution technologies are always on the horizon, thus closed captioning will continue to present new challenges in all aspects of the video production and distribution pipeline.

- Bottom Line: Once universal support is realized, the SMPTE 2052 specification will provide a long term solution for CEA-608/708 closed captioning for both broadcast and IP delivery, and other captioning and subtitling use cases
SMPTE Monthly Webcasts: 
Internet Captioning - Implications of the Multi-platform, Multi-Display Ecosystem

Q & A

About the Presenter
Jason Livingston is a developer and product manager with CPC Closed Captioning. He is well known for providing closed captioning software solutions to the industry. His recent projects include development of captioning software with speech recognition capabilities, and implementation of the latest SMPTE and CEA closed captioning standards.

Email: jason@cpcweb.com
Tel: 301-738-8487 x104

Host:
Joel E. Welch

SMPTE Monthly Webcast Sponsors

Thank you to our sponsors for their generous support of SMPTE and the SMPTE Professional Development Academy.

© 2013 • Powered by SMPTE Professional Development Academy • www.smpte.org