Building an IMSC Subtitle Archive

ENTERPRISE OPERATIONS

May 2018
Building a Subtitle Archive

Problems we’ve had with our archive

• Inconsistency
• Interoperability problems
• Constant re-manufacturing
Building a Subtitle Archive

We can all relate to these types of issues

- Poor sync
- Subtitles covering text
- Subtitles over matte
- Unreadable text
- Bad font choices
Building a Subtitle Archive

But Dave? How can we fix this?
Building a Subtitle Archive

Understand and control your end to end workflow

• Does your workflow begin with a stylized archive, or distribution?
• How many vendors author your subtitles?
• What tools will be used to create the subtitles?
• What tools will be used to transform the subtitles?
• Where will these be delivered?
• What compromises are you willing to accept?
Subtitle Workflow

Our end to end workflow

• We are constantly learning from our failures
• We want to create once, repurpose many
  • Archive
  • Stylized burn-in source
  • Distribution
  • HDR/SDR
• We work with numerous subtitle vendors
  • Subtitle style guide
  • Strict IMSC 1.0 specification
  • Involved technical onboarding process
• Our subtitles potentially go through various transformation or transcoding tools
• We’ve made compromises with our subtitle style and font style choices due to platform and device decoding inconsistencies
Subtitle Workflow

Streaming platforms have changed our approach

• Component localization has raised the bar for subtitles
  • Much tighter timing tolerances are expected
  • Higher quality transcriptions/translations are expected

• Component localization has lowered the bar for subtitles
  • Stylized subtitles are not required
  • Subtitle placement is simplified
  • Font size is variable based on playback device

• Certain IMSC 1.0 attributes are ignored or overwritten
Subtitle Workflow

Platform decoding behaviors

• Placement can change
• Visual style can change
• Font sizes can change

Whoa... Man, look, If this is about that poker game.
Subtitle Comprimes

How have we addressed these discrepancies?

• Our subtitle style is written so that platform modifications are minimized
  • Our default placement is bottom center
  • We add character identifiers to indicate the speaker when needed

• We still include an explicit visual style into our subtitles for consistency
  • Source for image based subtitles
  • Helps with QC on a mastering tool
  • Burning into picture is easy
  • Downstream platforms will eventually decode as described (hopefully!)
IMSC 1.0

How has IMSC 1.0 helped our workflows?

- Standardized file format for long-term archive
- Direct pass-through into our IMF Packages
- Direct delivery to our customers
- Timing issues with fractional frame rates can be minimized
- Complex font styles can be natively described
IMSC 1.0 Specification

We have standardized certain attributes – Time Expression

- Fractional frame rates are hard (sorry about that)
  - 23.976fps is SLOWER than real-time
- The typical media timebase syntax options are HH:MM:SS:FF, HH:MM:SS.ms, or frames
- Here’s the math for a 23.976fps time code value of 01:00:00:00:

\[
\text{HH:MM:SS:FF} \\
01:00:00:00 = 86400 \text{ frames} \\
86400 / (24000/1001) = 3603.6s \\
3603.6 = 01:00:03:14
\]

IMSC 1.0 XML Example:  ```<p begin="01:00:03:14"...```
IMSC 1.0 Specification

We have standardized certain attributes – Time Expression

• Fractional frame rates are hard (sorry about that)
  • 23.976fps is SLOWER than real-time
• The typical media timebase syntax options are HH:MM:SS:FF, HH:MM:SS.ms, or frames
• Here’s the math for a 23.976fps time code value of 01:00:00:00:

HH:MM:SS.ms

01:00:00:00 = 86400 frames
86400 / (24000/1001) = 3603.6s
3603.6 = 01:00:03.583

IMSC 1.0 XML Example:  <p begin="01:00:03.583"...
IMSC 1.0 Specification

We have standardized certain attributes – Time Expression

- Fractional frame rates are hard (sorry about that)
  - 23.976fps is SLOWER than real-time
- The typical media timebase syntax options are HH:MM:SS:FF, HH:MM:SS.ms, or frames
- Here’s the math for a 23.976fps time code value of 01:00:00:00:

frames

01:00:00:00 = 86400 frames

IMSC 1.0 XML Example:  
<p begin="86400f"...
IMSC 1.0 Specification

We have standardized certain attributes – Font Stylization

- fontFamily=""
- backgroundColor="black"
- color="white"
- textOutline=""
We have standardized certain attributes – Font Stylization

- fontFamily="proportionalSansSerif"
- backgroundColor="black"
- color="white"
- textOutline=""
IMSC 1.0 Specification

We have standardized certain attributes – Font Stylization

- fontFamily="proportionalSansSerif"
- backgroundColor="transparent"
- color="white"
- textOutline=""
IMSC 1.0 Specification

We have standardized certain attributes – Font Stylization

- fontFamily="proportionalSansSerif"
- backgroundColor="transparent"
- color="#DCDCDC"
- textOutline=""
IMSC 1.0 Specification

We have standardized certain attributes – Font Stylization

- fontFamily="proportionalSansSerif"
- backgroundColor="transparent"
- color="#DCDCDC"
- textOutline="black 4%"
IMSC 1.0 & HDR

Yes, it does actually work (with some effort)

- IMSC 1.0 has no separate luminance control. IMSC 1.1 adds this functionality
- Platforms must be educated on proper text based subtitle compositing
- Platforms have creative control over luminance level in their ecosystem
IMSC 1.0 & HDR

Yep, that was a good one.
Given the context of the image, the line in the text that stands out is: "Yep, that was a good one."
IMSC 1.0 & HDR

Yep, that was a good one.
Summary

Have a clear strategy

• Author once for archive and distribution needs
• Document and distribute clear specifications
• Work your distribution partners to ensure they can work with your assets
• Understand the end to end workflow, and control what you can
Questions?