What's new in the HbbTV Specification
“HbbTV 2.0.3 Explained”
<table>
<thead>
<tr>
<th>Informal Name</th>
<th>Formal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HbbTV 1.0</td>
<td>TS 102 796 V1.1.1</td>
</tr>
<tr>
<td>HbbTV 1.5</td>
<td>TS 102 796 V1.2.1</td>
</tr>
<tr>
<td>HbbTV 2.0</td>
<td>TS 102 796 V1.3.1</td>
</tr>
<tr>
<td>HbbTV 2.0.1</td>
<td>TS 102 796 V1.4.1</td>
</tr>
<tr>
<td>HbbTV 2.0.2</td>
<td>TS 102 796 V1.5.1</td>
</tr>
<tr>
<td>HbbTV 2.0.3</td>
<td>TS 102 796 V1.6.1 (tbc)</td>
</tr>
<tr>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>
3 Elements to HbbTV 2.0.3

• Errata to HbbTV 2.0.1/2
  – Fixing bugs in the spec

• Updates to existing features
  – Goal for 2.0.3 was “low hanging fruit” that are easy to specify & test
    • No big new features – these were deferred to the next iteration
  – Critical updates already widely supported in practice
  – One small new feature

• Removing unused and replaced features
  – Cannot keep adding features and never removing anything
What are Errata?

• Fixes to bugs in the spec
  – May be cosmetic
    • Punctuation, cross-references, …
  – Might be language that’s unclear or hard to understand
  – May be ambiguities
    • Something that can genuinely be interpreted in more than one way
  – May be conflicts & inconsistencies
    • Statements that actually say different things
  – Things that are hard or even impossible to implement in the real world

• HbbTV publishes >1 document for each errata release
  – A list of the changes to the specification(s)
  – A version of the specification(s) with the errata integrated & changes tracked

• All HbbTV errata have an issue number
  – Issue numbers can be used to cross-reference between the two documents
  – HbbTV members can use the number to lookup the discussion in our issue tracking system

• 2 recent errata releases; July and October
  – 2.0.3 started as a fork of 2.0.2 with July 2020 errata integrated
  – Errata found / resolved between July and October are included in the October release of errata for 2.0.1/2 and in 2.0.3
• Design of HTML5 video element doesn’t take account of media devices using hardware to decode media

• Many, many small issues and corner cases result
  – #10195: media decoders and the sought event
  – #10181: Releasing resources from HTML media elements
  – #9623: language in clause 9.6.2 about hardware video resource management and HTML5 video elements
  – #9481: Potential conflict between HbbTV 9.6.2 and HTML5 re HTML5 load() method taking video & audio decoders

• Related work in W3C Media and Entertainment Interest Group
  – https://github.com/w3c/me-media-integration-guidelines/
  – W3C members should take a look at the issues & perhaps contribute
Errata to HbbTV 2.0.1/2 (2)

- Media synchronisation requirements may be unrealistic for some implementers
  - #9325: Unreasonably demanding a/v sync timing requirement
    • Relaxed from -10ms/+10ms to -35ms/+50ms
  - #10435: Unreasonable demanding of synchronization between A/V and subtitles
    • Tests assume 40ms, market expects frame accuracy at scene cuts, may not be achievable on some hardware

- Media synchronisation (video via broadcast and audio via broadband)
  - #8810: behaviour of multi-stream sync API at times when no content exists in a slave media synchroniser
  - #10719: stopping multi-stream sync and disposing of a MediaSynchroniser object
  - #10722: successful completion of initMediaSynchroniser and addMediaObject
Video via broadcast and audio via broadband

Alternate audio could be other languages or audio for accessibility e.g. audio description, clean audio, ....
• Compatibility with modern soft text input (virtual keyboards)
  – #10007: Section 10.2.1: incompatibility with modern soft input (virtual keyboards) that operate on words or phrases
    • Turns out that this also enables some forms of voice input
• DASH
  – #10447: Errata to DVB-DASH
    • Update to newest version to get bug fixes – optional features remain optional, features added by DVB are all optional
  – #9315: DASH - MPD events
    • SCTE-35 ad insertion events crash some HbbTV implementations
• Future evolution and maintainability of the XML capabilities mechanism
  – #9487: Error in XML capabilities example and XSD
Updated Features – Web Standards

- HbbTV 2.0.0/1/2 all based on a 2013 selection of web standards
  - Open IPTV Forum "Web Standards TV Profile"
    - A few select web standards added by HbbTV but nothing systematic
  - The web has moved on
- HbbTV 2.0.3 is based on a 2018 selection of web standards
  - CTA WAVE "Web Media API Snapshot"
    - Standards implemented by all desktop browsers in 2018 – including some not previously in HbbTV
- Why 2018?
  - Why not 2020?
    - To allow time for code to be ported and optimised for constrained systems
  - Why not 2016?
    - Public disclosure of security bugs in desktop browsers means that shipping TVs based on old browsers may be unwise
- In practice, HbbTV 2.0.1/2 implementations will be based on a browser more recent than 2013
  - But perhaps not as recent as 2018
  - Moving to a new version of Chrome/Webkit will increase the product development cost
  - The more someone optimises a port of Chrome/Webkit for TV, the longer they want to keep using it
  - New devices continuing to use very out of date browsers is a major concern for some
    - Remember the HTML5 engine in an HbbTV TV isn’t normally updated unlike Chrome or Safari in a mobile
API for media playback widely used for video content in the web
  - Complex player logic shifted from HbbTV implementer to JavaScript library in the app
    - App handles loading data & uses MSE to pass it to the media decoders in the terminal
    - MSE can be used with DASH, HLS and probably any HTTP-based protocol/format that can be implemented as a JavaScript library
  - Content / app providers may chose among a number of MSE player libraries
    - Open source libraries for DASH
      - `dash.js`
      - `Shaka Player`
    - Also HLS open source libraries and commercial offerings for DASH, HLS and others
  - Some HbbTV terminals have supported MSE for years
    - HbbTV apps using MSE deployed in Germany starting in 2018
      - Fraunhofer FOKUS “Kumpel-Tag mit Andy” for WDR
    - MSE required for HbbTV targeted advertising
      - App loads advert into memory & can play it with guarantees of no pausing / stalling due to the network
360° HbbTV Application: “Kumpel-Tag mit Andy”

- Limit the range to new HbbTV devices with support of Media Source Extension (MSE) API through whitelisting

- Testing and customizing filtering

- Red-Button Teaser leads via the WDR Start bar to the landing page

- Landing page with Animated GIF explains 360° navigation

- 80% of video users navigated to different perspectives
W3C Media Source Extensions

• Key advantages of MSE
  – Reduction in DASH interop problems as content/app provider can choose a DASH player known to work with their encoder/packager provider
    • Potential saving for content/app providers, their suppliers and manufacturers
  – JavaScript libraries can evolve as streaming media protocols evolve without needing software updates on HbbTV terminals
  – Consistency with the way commercial media playback is done in the web, re-use of tools, libraries, apps, expertise

• Some known limitations
  – Some low-end terminals may be able to play UHD via the native DASH player but not have enough processor capacity to play UHD via MSE
  – Using generic web APIs to load the data (XHR, fetch) brings security requirements that may not be useful with commercial media
    • Mixed content -> use of TLS even for DRM encrypted content, extra network traffic due to CORS
  – MSE implementations in chrome/webkit don’t support some advanced media features & integrators don’t want to repeatedly modify code with each upstream release
    • e.g. exposing NGA Preselections as HTML5 AudioTrack objects & selecting between them
    • Chrome in general doesn’t support HEVC via MSE
Updated Features – Web Standards

- Other web standards not previously required (that may be there anyway)
  - Service workers
    - Enable more responsive and adaptable apps
  - Recent web security specifications
  - HTTP/2
  - navigator.cookieEnabled
    - Querying if persistent storage of cookies & web storage is disabled
- TLS updated to version 1.3
  - Not in all 2018 browsers but being rolled out very quickly
  - https://www.caniuse.com/tls1-3
Version 1.3 (the latest one) of the Transport Layer Security (TLS) protocol. Removes weaker elliptic curves and hash functions.

<table>
<thead>
<tr>
<th>Feature: TLS 1.3</th>
<th>Usage</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.3</td>
<td></td>
<td>88.94%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ 2.44%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>91.38%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>UC Browser for Android</th>
<th>Samsung Internet</th>
<th>QQ Browser</th>
<th>Baidu Browser</th>
<th>KaIoS Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2-50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12-18</td>
<td>60-62</td>
<td>70-85</td>
<td>4-53</td>
<td>54-69</td>
<td>3.1-12</td>
<td>54-56</td>
<td>3.2-12.1</td>
<td>2.1-4.4.4</td>
<td>12-12.1</td>
<td>4-9.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td></td>
<td>79-85</td>
<td>63-81</td>
<td>70-85</td>
<td>2.1-13.1</td>
<td>57-71</td>
<td>12.2-13.7</td>
<td>14</td>
<td>81</td>
<td>63-81</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>86</td>
<td>82</td>
<td>86</td>
<td>14</td>
<td>72</td>
<td>14</td>
<td>86</td>
<td>82</td>
<td>12.12</td>
<td>12.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Current aligned** | **Usage relative** | **Date relative** | **Filtered** | **All** |
Updated Features – OTT Streaming; CMAF

• DASH builds on the ISOBMFF (.mp4) file / container format
  – This is a complex spec with many options
  – DASH doesn’t itself define a clear profile of the format

• CMAF
  – Comes from an Apple / Microsoft co-operation to switch HLS from transport stream to .mp4
  – Both DASH and HLS manifests should be able to refer to the same CMAF media segments
  – Started out as a minimum basic set of features that should work everywhere

• CMAF and HbbTV
  – CMAF container format is a subset/profile of ISOBMFF so in theory CMAF content should just work on HbbTV
    • Not possible to prove this but I’ve asked many times & nobody could think of anything in the CMAF container format that isn’t effectively required for DASH
  – Unlike DASH, CMAF also defines codec profiles
    • May be issues here if content uses unusual picture aspect ratios, resolutions, (etc) as DVB-DASH is (much?) more constrained

• DASH-IF validator can be used to validate CMAF compatibility as well as (DVB-)DASH compatibility
Updated Features – OTT Streaming
Low latency live / linear services

- Low latency is very fashionable right now but also complex
  - Several parts to it & several options for doing it
- Start-up delay
  - Time until media is visible / audible
  - Applicable to all linear content
- Catching up & closely following the live edge
  - Mostly applicable to really live content
    - e.g. ensuring that people watching football see a goal on TV before seeing tweets about it
    - Includes catching up with the live edge if the player starts behind or falls behind
- 2 obvious options for LL (see DVB-DASH 11.18.1)
  - Small segments – e.g. 1s instead of 4s or 8s
    - In practice should work everywhere but brings bandwidth & CDN overhead
  - Chunked content (aka multiple moof/mdat boxes per DASH segment)
    - Beginning of a DASH segment can be downloaded before the end has been encoded or even exists
    - In theory should work everywhere but may not in practice on older devices – testing required
- HbbTV 2.0.3 supports low latency via MSE
  - Native DASH player should play both options for LL content in a backwards compatible mode but not required to catch up & follow the live edge

HbbTV Association | Copyright © HbbTV
Chunked Content & HbbTV DASH/DRM Reference App

Updated Features – OTT Streaming
Querying CBCS Encryption

• Industry is moving to adopt Apple’s flavour of AES – CBCS - instead of CENC as previously used
  – Widevine already moved
  – PlayReady 4.0 supports Apple flavour of CBCS
• CBCS is one element of enabling content to be encoded, packaged & encrypted once for many different devices
  – CMAF is another element of this
  – Obviously care still needed with codec choices, resolutions, …
• HbbTV 2.0.3 allows apps to query which AES encryption modes are supported
The One New Feature

- Querying physical screen size
  - Extension to HbbTV XML capabilities mechanism

Terminals with a built-in or HDMI-connected display shall include one or more elements of the following form to describe the size of display:

```xml
<display_size width="w" height="h" measurement_type="t"/>
```

where w and h are integer values describing the horizontal width and vertical height of the display respectively, both in units of centimeters, and t is a string taking one of the following values:

- "built-in", where the display forms an integral part of the terminal. In this case, the width, w, and height, h, shall be accurate to within 5cm.
- "hdmi-accurate", where the display is connected by HDMI and the width, w, and height, h, are reported by the display as being accurate to within 5cm or less.
- "hdmi-other", where the display is connected by HDMI and the width, w, and height, h, are not reported as accurate to within 5cm.
Unused and Replaced Features

• A spring cleaning of features that are not used in the real world or which have been replaced
• Some features have been removed immediately
  – Mostly gone completely, one feature moved to OpApp
• Some features are at risk ("deprecated")
  – At risk of being moved to the OpApp spec
  – At risk of being removed in the next specification release
  – At risk of being removed in the further future
Features already removed from 2.0.3

Remember these are all features not used in the real world

- **CI+ host player mode**
  - DASH player in TV/STB uses DRM system in separate hardware module (USB/PCMCIA)

- **HbbTV app launching an app on a phone (in that direction only)**
  - Design relied on TV/STB manufacturers including support for it in a mobile app that consumers install
    - Larger manufacturers didn’t include this in their apps & smaller ones never had apps in the 1st place
  - A mobile app launching app on a TV/STB (the opposite direction) remains mandatory

- **Teletext Subtitles in OTT content**
  - It seemed logical in 2009 & there wasn’t a lot else for subtitles at the time

- **3 aspects of media sync**
  - Use of A/V control object in media sync
    - Why add functionality to the 10 year-old API? apps can just use the HTML5 video element
    - Media sync tests in the test suite using the A/V control object are being re-worked or dropped
  - 2 never implemented options
    - SYNC_SLAVE mode
    - Sync buffer

- **Moved to the OpApp spec**
  - **CI+ CICAM player mode**
    - Media player (DASH, HLS, other) and DRM system together in separate hardware module (USB/PCMCIA)
Features at risk of being moved or removed

- **Candidate to move to the OpApp spec**
  - Push VoD & download manager
    - Some prototype implementations but not known to be deployed in the real world
- **Re-consider in next requirements cycle**
  - (Local) PVR
    - Has been implemented, in whole by some & just the subset relevant to timeshift by others
    - Test cases exist but were never reviewed - volunteers needed to review them
- **Depends on privacy considerations**
  - File system acceleration
    - HbbTV Privacy Task Force to evaluate if this is still needed based on evolving privacy landscape

- **Candidates to be removed at some time in the future**
  - A/V Control object
    - Replaced by HTML5 video element
  - oipfDRMAgent object
    - Replaced by Encrypted Media Extensions
  - As these are used by many apps today, plenty of notice will be given!
  - May be possible to replace A/V Control Object with a web “Polyfill” if interest
  - No new functionality to be added using A/V control object
    - Media sync functionality added in HbbTV 2 has been removed in 2.0.3
Testing HbbTV 2.0.3

- Up to 83 new HbbTV test cases
  - 35 MSE
    - Also some tests for MSE in the targeted advertising option become mandatory for 2.0.3
  - 18 for new version of DVB-DASH (also applicable for 2.0.1/2)
  - 14 others specific to 2.0.3
  - 16 for 2.0.1/2/3 (e.g. multi-stream sync accessibility use-cases)
  - These should be included in July 2021 HbbTV test suite release targeting 2022 HbbTV products
    - Hopefully early access versions can be included in March 2021 test suite release

- Web standards
  - Historically HbbTV has not tested these
    - Hope to include a small sample of W3C Web Platform Tests to confirm that recent APIs were implemented at all
    - Never a high enough priority
  - CTA WAVE project has done a lot of work to make W3C Web Platform Tests more suitable for Smart TVs
    - [https://github.com/cta-wave/WMAS](https://github.com/cta-wave/WMAS)
  - Details still be worked out
Looking Forwards

- **What comes after 2.0.3?**
- **New requirements just finalised in HbbTV, examples include:**
  - Co-existence of HbbTV apps with accessibility features offered by TV / STB
    - e.g. screen magnifier, feedback on interaction, high contrast UI, dialogue enhancement for NGA
    - HbbTV will **not** make any of these mandatory, just permit apps to discover what’s there, work with it & avoid conflicts
    - Looking forwards to the implementation of the European Accessibility Act in 2022-2025
  - Co-existence of HbbTV apps with voice assistants offered by TV / STB
    - Someone using voice input with the TV / STB UI should not have to switch back to classic remote control for HbbTV apps
  - Support for HbbTV “red button” apps on DVB-I services
- **Will it be called 2.0.4?**
  - No decision on the informal name yet
Some Opportunities to Contribute

- HbbTV 2.0.4 (tbc)
  - See previous slide; help with Alexa / Google Assistant particularly welcome
- Unit tests for HbbTV 2.0.3 and errata to 2.0.1/2
  - Help running early access versions of 2.0.3 unit tests from Feb/Mar 2021 onwards
  - Contribute unit tests to fill gaps (“Unplanned submissions”)
  - Expertise on W3C Web Platform Tests to help include them in HbbTV
- Report real-world inter-operability problems
  - Problems you can reproduce on HbbTV TVs/STBs from *multiple* manufacturers (ideally 2020 models)
- Fork the DASH-DRM reference app, do something interesting / relevant to a wider audience & contribute it back to HbbTV
- Help with creating a web polyfill mapping CE-HTML A/V object to HTML5 video element

1 HbbTV membership required
Thank You